DEVELOPMENT OF A SIX DEGREE OF FREEDOM MOTION SIMULATION MODEL FOR USE IN SUBMARINE DESIGN ANALYSIS.

John Thomas Hammond

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DEVELOPMENT OF A SIX DEGREE OF FREEDOM MOTION SIMULATION MODEL FOR USE IN SUBMARINE DESIGN ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED THESIS
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) HAMMOND, JOHN T.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS MASS. INST. OF TECHNOLOGY		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS CODE 031 NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA, 93940		12. REPORT DATE MAY 78
		13. NUMBER OF PAGES 87
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

MOTION SIMULATION; SUBMARINE DESIGN; SIX DEGREE OF FREEDOM

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE REVERSE.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

UNCLASS
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DEVELOPMENT OF A SIX DEGREE OF FREEDOM MOTION SIMULATION MODEL FOR USE IN SUBMARINE DESIGN ANALYSIS

by

John Thomas Hammond

Submitted to the Department of Ocean Engineering on May 12, 1978 in partial fulfillment of the requirements for the Degrees of Ocean Engineer and Master of Science in Naval Architecture and Marine Engineering.

ABSTRACT

A computer mathematical model that will simulate the six degree of freedom motion of a submarine has been developed. Hydrodynamic coefficients obtained from the testing of physical models enable the user to accurately simulate specific submarine designs. The simulation model can be used as a design tool to study and predict submarine dynamic response. Complete three dimensional motion is allowed for the submarine while constraints in powering, rudder deflection, dive plane angle, and response time can be selected by the user. Outputs include a chronological history of all linear and angular velocities and accelerations, the instantaneous angle of deflection of each control surface, and the trajectory of the submarine center of gravity.

Thesis Supervisor: Martin A. Abkowitz

Title: Professor of Ocean Engineering

DEVELOPMENT OF A SIX DEGREE OF FREEDOM MOTION SIMULATION MODEL FOR USE IN SUBMARINE DESIGN ANALYSIS

by

John Thomas Hammond
Lieutenant, United States Navy

B.S.M.E., University of Washington
(1971)

Submitted in partial fulfillment
of the requirements for the
degrees of

Ocean Engineer
and
Master of Science
in Naval Architecture
and Marine Engineering

at the

Massachusetts Institute of Technology

May, 1978

DEVELOPMENT OF A SIX DEGREE OF FREEDOM
MOTION SIMULATION MODEL FOR USE IN
SUBMARINE DESIGN ANALYSIS

by

John Thomas Hammond

Submitted to the Department of Ocean Engineering on May 12, 1978 in partial fulfillment of the requirements for the Degrees of Ocean Engineer and Master of Science in Naval Architecture and Marine Engineering.

ABSTRACT

A computer mathematical model that will simulate the six degree of freedom motion of a submarine has been developed. Hydrodynamic coefficients obtained from the testing of physical models enable the user to accurately simulate specific submarine designs. The simulation model can be used as a design tool to study and predict submarine dynamic response. Complete three dimensional motion is allowed for the submarine while constraints in powering, rudder deflection, dive plane angle, and response time can be selected by the user. Outputs include a chronological history of all linear and angular velocities and accelerations, the instantaneous angle of deflection of each control surface, and the trajectory of the submarine center of gravity.

Thesis Supervisor: Martin A. Abkowitz

Title: Professor of Ocean Engineering

ACKNOWLEDGEMENTS

This thesis was prepared under the auspices of the Department of Ocean Engineering and Professor Martin A. Abkowitz. My sincerest thanks to Professor Abkowitz for his expert guidance in developing the simulation model used in this thesis. I would also like to express special thanks to my wife Robin for her loving support during the months of thesis preparation and for her endless hours of expert typing.

J.T.H.

MAY 1978
CAMBRIDGE, MASSACHUSETTS

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I - INTRODUCTION

I.1 BACKGROUND

During the several years spent in designing a submarine, a great deal of effort is expended trying to improve the design so that the best possible vessel is eventually constructed. During the design phases the designer will frequently examine problems that were found in older submarines and try to determine the cause of each problem so that they can be eliminated in the new design. At several stages during the design, computer models are used to assist in such things as structural design, equilibrium calculations, and internal arrangement. As the design progresses, physical models of the proposed vessel are built and tested in a towing tank to measure the hydrodynamic coefficients. Finally, when it is possible to deal with the vessel as a whole, a dynamic analysis is conducted to gauge how the submarine will perform in an underwater environment when all six degrees of freedom are available.

While it is possible to gather dynamic information, such as the hydrodynamic coefficients, from physical model tests, the models are necessarily too restricted in their motion to permit a complete

analysis. It is necessary to construct a mathematical model of the submarine to simulate the submarine's motion and thereby obtain enough data for a complete analysis. The simulation model will use the hydrodynamic coefficients obtained from the physical model as a basis for the simulation. A properly constructed model will permit the designer to simulate any conceivable maneuver and gauge the submarine's response. The simulation model then becomes a tool to assist in improving the design and achieving the best possible vessel. After the submarine is designed, the model can continue to be useful by assisting in the evaluation of operating procedures. The model can be placed in any maneuvering situation without hazard to crew or vessel. This is especially useful in evaluating casualty situations. The model can also be used to estimate the effect of proposed design changes to the submarine. For instance, the model can show the effect of changing the maximum deflection of a control surface or its rate of operation.

All of the few six degree of freedom submarine simulation models in existence are too expensive for daily use in the design office. Generally, the models are complicated to use and sometimes difficult to obtain. These problems have created the need for a program that is both inexpensive enough to permit daily use in the design office and simple enough so that anyone can use it. This need has formed the motivation for this thesis.

I.2 MODEL DEVELOPMENT

A mathematical model must use Newton's Law of Motion and the

differential equations resulting from a dynamic analysis of the submarine. A submarine has movable appendages, so these must also be accounted for in the model. Newton's Law of Motion can be expressed as:

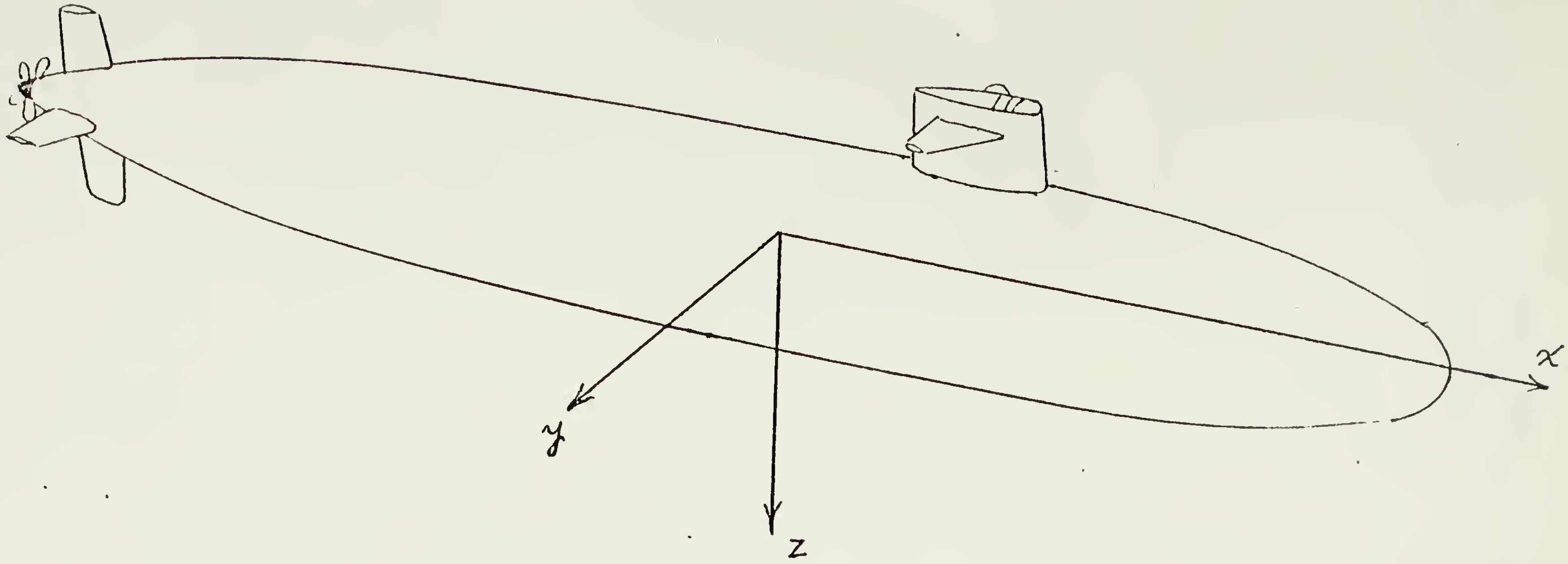
$$\text{(1)} \qquad \text{Force} = \frac{d}{dt}\ (\text{momentum})$$

$$\text{(2)} \qquad \text{Moment} = \frac{d}{dt}\ (\text{angular momentum})$$

If a submarine with a coordinate system such as shown in Figure 1 is used, then equation (1) can be applied along each of the three axes. Similarly, equation (2) can be applied around each axis. This gives a total of six equations to represent the six degrees of freedom of the submarine. These six equations of motion are well-known [1] so they are not developed in detail here. They are, however, included for reference in Appendix A.

While the equations of motion for the submarine form the heart of the model, they are insufficient by themselves. On an actual submarine, the officer of the deck orders the rudder or dive planes moved in order to maneuver the ship. The routines will normally sense where the ship currently is and where it is supposed to be and then move the appendages in the appropriate direction just as the deck officer would do. The six equations of motion together with the appendage control subroutines constitute the vital components of the simulation model. A main program is necessary to coordinate the input, output, and to control the action of the subroutines.

The components of the model are discussed in detail in Chapter II of this thesis. The model is written as a computer program whose features are discussed in Chapter III. The final chapter will



Local Submarine Coordinate System

Figure 1

discuss the model tests and their results. A listing of the program will be provided in Appendix B.

CHAPTER II - THE SIMULATION MODEL

II.1 GENERAL CAPABILITIES

The simulation model developed in this thesis will provide trajectory information for routine submarine maneuvers such as turning or changing depth. The trajectory, which is referenced to a fixed coordinate system, is computed as a function of time. As the trajectory of the model is developed, the velocities and accelerations are calculated and stored for output to the user. Both angular and linear velocities and accelerations are provided. The angle of deflection of each control surface appendage is computed for each step of the entire trajectory or maneuver.

The rate of deflection of each control surface and its angle of maximum deflection are specified by the user. Any of the control surfaces can be "jammed" by specifying a particular angle of deflection. The model can be initially placed on any course and depth and then ordered to come to any new course and depth. The user can select the initial speed and the appropriate thrust coefficients to cause a change in speed.

II.2 COORDINATE SYSTEMS

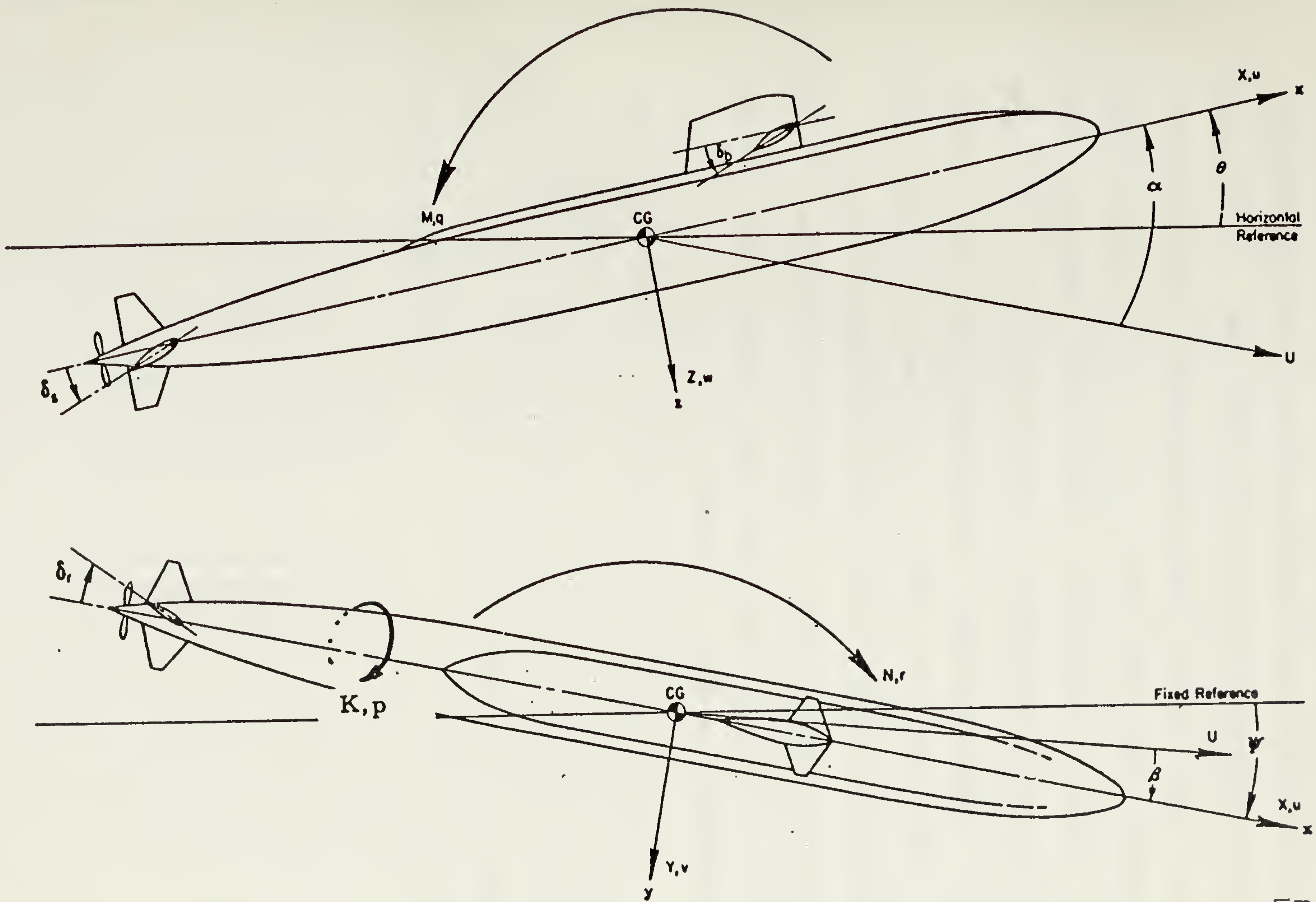
The coordinate system plan used in the simulation model is based on [1]. The model uses two orthogonal coordinate systems; one remaining fixed at the water surface while the other travels with the submarine to act as a local reference system. The fixed system, designated x_0, y_0, z_0, is related to the moving system, designated x, y, z, through the angles ϕ, θ, ψ as illustrated in Figure 2. Quantities measured in the moving system can be referenced to the fixed system by using the transformation [2] given in Appendix A.

The origin of the moving system is at the center of gravity of the submarine. This has the advantage that only the principal moments of inertia, I_x, I_y, I_z, are non-zero (i.e.: $I_{xy}=I_{yz}=I_{xz}=0$). It has the additional advantage that the equations of motion in [1] also use this reference point for the x, y, z system. It would have been possible to use the centerline of the submarine for the origin. This would have the advantage of making better use of the symmetry of the vessel, however it would have the disadvantage of having to correct both the equations of motion and some of the hydrodynamic coefficients for the new origin.

Appendage movements are measured with respect to the moving coordinate system. All velocities and accelerations are measured along the axes of the moving system. This is also shown in Figure 2.

II.3 EQUATIONS OF MOTION

The general nature of the six degree-of freedom model requires the use of the six equations in their non-linear form. A large



Relationship of Axes, Angles, Velocities, Forces, & Moments [1]

Figure 2

number of hydrodynamic coefficients are necessary for this model. It would be desirable to derive each coefficient from existing hydrodynamic theory. Some coefficients have been obtained for bodies of revolution or other mathematically amenable shapes; however, when appendages such as control surfaces, fairwaters, and propellers are taken into account, the accuracy of the theoretical values is brought into question. For this reason it is standard practice to obtain the numerical value of the coefficients by experimental means. This usually entails the towing and measurement of physical models.

Once the hydrodynamic coefficients are known, the equations can be solved for the accelerations. The six equations of motion must be written such that the highest order derivatives, namely $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{p}$, $\dot{q}$, $\dot{r}$ and their coefficients, appear on the left hand side of the equation. This gives each equation a form like:

$$a_{i,j}\,\dot{u} + a_{i,j+1}\,\dot{v} + a_{i,j+2}\,\dot{w} + a_{i,j+3}\,\dot{p} + a_{i,j+4}\,\dot{q} + a_{i,j+5}\,\dot{r}$$
$$= \quad f_i\,(u, v, w, p, q, r, \phi, \theta, \psi, \delta_r, \delta_b, \delta_s, l, m, \ldots)$$

The left hand side is placed in a matrix format. The six equations are then represented as:

$$\begin{bmatrix} a_{i,j} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}$$

The method of solution is an iterative technique in which initial values are used in the right hand side functions and the six accelerations are solved for on the left side. The accelerations are then used to update the right side functions. A small time increment is made and the accelerations are again solved by using the latest value for the right side functions.

With each iteration the accelerations are used in a Taylor series expansion to calculate the velocities. The calculations have the form:

$$\begin{Bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{Bmatrix}(t+\Delta t) = \begin{Bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{Bmatrix}(t) + \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} \cdot (\Delta t)$$

The pitch, roll, and yaw angles are calculated in a similar manner using the angular velocity and acceleration.

$$\begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}(t+\Delta t) = \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}(t) + \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \cdot (\Delta t) + \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} \cdot \left(\frac{\Delta t^2}{2}\right)$$

In order to plot the trajectory of the submarine, a coordinate system transformation is performed to obtain the linear velocities $\dot{x}_o$, $\dot{y}_o$, $\dot{z}_o$ in the fixed coordinate system. The velocities are used in a Taylor series expansion to obtain the position of the center of gravity of the submarine in the x_o, y_o, z_o system.

$$\begin{Bmatrix} x_o \\ y_o \\ z_o \end{Bmatrix} (t + \Delta t) = \begin{Bmatrix} x_o \\ y_o \\ z_o \end{Bmatrix} (t) + \begin{Bmatrix} \dot{x}_o \\ \dot{y}_o \\ \dot{z}_o \end{Bmatrix} \cdot (\Delta t)$$

When the position of the submarine is calculated the time is advanced one increment and another iteration begins.

The iteration loop will continue to operate until some test criteria are met. For instance, if the model is performing a dive, then the model tests the value of z_o to see if the model is at the appropriate depth. A test is also made of the pitch angle to see if it is within some specified range of zero. Lastly a check is made to ensure that all of the dive planes are at zero deflection.

II.4 CONTROL SURFACES

The motion of the model is controlled by the movement of the control surfaces. The control surfaces include the rudder, the stern planes, and the sail planes. For analysis of submarine designs the movement of the control surfaces can be governed by an automatic control system. In the model each control surface is provided with its own automatic control system. The principle of operation for all of the automatic control systems is the same. In each case the deflection of the control surface is made proportional to an error signal and the rate of change of the error. The sail planes, for instance, control the depth of the model. The calculated deflection of the planes is proportional to the depth error and the rate of change of depth.

$$\delta_c = K_1 \text{ (present depth - ordered depth)}$$
$$+ K_2 \text{ (rate of change of depth)}$$

A large error will initially cause a large deflection, but as the rate of change increases, the deflection will decreáse until some moderate rate is achieved. Since movement of the sailplanes does not have any significant effect on any motion other than depth, its control system uses only the variables associated with depth.

The stern planes control pitch angle as well as depth. The control system accounts for this by using the pitch angle, θ, and the rate of change of θ.

$$\begin{aligned} \delta_c &= K_3 \text{ (present depth - ordered depth)} \\ &+ K_4 \text{ (rate of change of depth)} \\ &+ K_5 \text{ (pitch angle)} \\ &+ K_6 \text{ (rate of change of pitch angle)} \end{aligned}$$

The K's in the control systems are proportionality constants known as gain. The value of the gain determines the sensitivity and response of the control system. Since the error signal continually varies during a maneuver, it is best to keep the gain low. It is desirable to have just enough gain on the error signal to get the control system moving and keep it moving in the right direction. The gain on the rate signals should be much stronger to dampen out the oscillations caused by response to the error signal.

The control surfaces move at a rate specified by the user. Whenever the calculated deflection differs from the present deflection, the control surface will move toward the calculated value at the specified rate.

CHAPTER THREE - PROGRAM USER'S GUIDE

III.1 GENERAL FEATURES

The simulation model consists of a main program and four subroutines. The four subroutines are named FUNC, RUDDER, DEPTH, and STERN. The main program reads the input data, changes units, initializes values, and prints the output. The main program performs the Taylor series expansions, the coordinate system transforms, and provides the logical statements for calling the subroutines. On the first iteration, when time equals zero, no subroutines are called; the output from this first iteration is then a statement of the initial conditions of the problem. Each succeeding iteration will call subroutine FUNC to calculate the new position of the model. Calls to the control surface subroutines are made on the basis of need, with no subroutine being called more than once in the iteration. The decision on whether or not to call RUDDER, DEPTH, or STERN is contained in a series of logical IF statements in the main program.

III.2 SUBROUTINE FUNC

Subroutine FUNC contains the six equations of motion. The

input parameters to FUNC consist of the hydrodynamic coefficients, the model velocities, the orientation in space, all of the ship's characteristics such as length and moments of inertia, the deflection of each control surface, and the propulsive coefficients. The subroutine returns the value of the six accelerations: UDT, VDT, WDT, PDT, QDT, RDT. The input and output for FUNC are all contained in COMMON /FOUR/ and COMMON /FIVE/. The solution to the matrix equation in subroutine FUNC is made possible by a call to the library function LEQT1F [3]. LEQT1F performs a Gaussian reduction for the matrix equation. Any similar Gaussian reduction could be substituted if LEQT1F is not available. An explanation of the parameters used in the call to LEQT1F is found in Appendix B.

III.3 SUBROUTINE RUDDER

Subroutine RUDDER controls all horizontal motion for the model. The input parameters include: K9, K10, T5, T6, T11, T12, T17, T18, TLAGR, RRATE, RUDAMT, COURSE, DELR, DELT, R, PSI. The subroutine returns a new value for DELR, the rudder deflection, based on a calculation using its present and ordered headings. The inputs are all contained in COMMON /THREE/ and COMMON /FIVE/. Subroutine RUDDER is called whenever the model is not on the desired course or when the rudder is deflected.

III.4 SUBROUTINE DEPTH

Subroutine DEPTH controls the forward set of dive planes. It can control either bow planes or sail planes depending on what the

submarine is fitted with. This subroutine is sensitive to the depth error and the rate of change of depth. Subroutine DEPTH will move the forward planes in the direction necessary to bring the depth error to zero. As input parameters, the subroutine uses: K1, K2 T1, T2, T13, T14, T15, T16, TLAGB, DELT, DIFF, ADIFF, ZDT, ATHETA, MAXANG, DCRIT. The subroutine returns a new value for DELB, the bow plane deflection, after each call.

If MAXANG, the maximum dive/ascent angle, has been exceeded, then DEPTH will return a diagnostic write statement. The user may specity MAXANG, the maximum pitch angle for the model. Whenever MAXANG is exceeded the diving planes are moved to reduce the pitch angle and a diagnostic signal is generated from within subroutine DEPTH. The diagnostic will say "Maximum dive/ascent angle exceeded at time ____. Standard fixup taken." Since the dive planes only begin to react when MAXANG is exceeded, the submarine will overshoot the angle before a reduction in the angle actually occurs. The diving planes will continue to operate until the pitch angle is less than MAXANG.

DEPTH makes only a minimal attempt to control the pitch angle of the model; it only warns the user when the angle is exceeded, and then moves the bow planes so that the pitch angle does not grow larger. The principal purpose of DEPTH is to provide depth control; it does this in conjunction with subroutine STERN.

III.5 SUBROUTINE STERN

Subroutine STERN controls the movement of the stern planes to

achieve the desired depth and pitch angle. The movement of the planes is sensitive to the depth error, the rate of change of depth, the pitch angle, and the rate of change of pitch. The stern planes will move to bring the depth error to zero and the pitch angle to zero. As input parameters the subroutine uses: K5, K6, K7, K8, T3, T7, T8, T10, TLAGS, STERAT, STERMX, THETA, Q, DELS, DIFF, ADIFF, ZDT, ATHETA, MAXANG, NOPICH, DCRIT. The subroutine returns a new value for the stern plane deflection, DELS. The input and output parameters are all contained in COMMON /ONE/, COMMON /FIVE/, and COMMON /SIX/. STERN is called to achieve a depth change or to correct the pitch angle.

III.6 <u>RANGE VARIABLES</u>

Since it is not usually possible to bring a computer simulation model to a precise depth or angle, it is necessary to define ranges around the desired depth or angle which will be acceptable to the user. For example, if the submarine were to make a depth change of 500 feet, the dive may be considered complete if the model settled within ten feet of the desired depth. The range is specified by the user to enable him to achieve whatever precision is desired. Within this range the main program will not make a call to the control surface subroutine; since no call is made, the control surfaces cannot be moved. In the program, the variable name for the depth range is NODIFF; it is usually set at $\pm$ 5 or $\pm$ 10 feet. Subroutine DEPTH will be called whenever the model is off of its ordered depth by more than NODIFF. The range about zero pitch angle is given the name NOPICH;

it is usually set at ± 1 degree. STERN is called if the pitch angle is greater than NOPICH. The variable name for achieving the proper course is ONCRS; it is usually set at ± 1 degree. RUDDER will be called whenever the model is off course by more than ONCRS.

An additional range variable is used during diving or surfacing. When the model moves within some critical range of the desired depth, it is time to begin moving the dive planes to zero angle and let the vessel glide into the desired depth. This maneuver prevents oscillation of the dive planes as the error signal and the rate signal both become small. This also ensures that the planes are at or near zero angle by the time the desired depth is reached. The name of this variable is DCRIT; it is usually set at 50, 75 or 100 feet depending on the speed of the submarine. STERN will be called whenever the model is off its ordered depth by more than DCRIT.

III.7 PROGRAM PARAMETERS

The iterative nature of the program relies on a time increment being made after each step. The size of the time increment is optional; the smaller the increment the more accurate the calculations. In selecting the size of the time increment, one must bear in mind the length of time required to complete the maneuver and the storage capacity of the program. Due to the quantity of information that is calculated by the program, it is practical to print only half of the data at one time. The other half of the data is stored in the array STOW. This data is then printed when the maneuver is complete. STOW has the capacity to hold the data resulting from 600 iterations.

A time check is incorporated in the main program to enable the user to limit the number of iterations. Achieving a number of iterations equal to the variable ICNT will cause the program to stop the present maneuver, print out all data, and see if another maneuver is desired. ICNT should not be made larger than 600 so that the available storage space is not exceeded.

Each control surface subroutine has a time lag scheme which senses the initial command to the control surface and prevents immediate action. A time lag occurs each time the direction of movement is changed. A different time lag may be specified for each control surface.

The variable INDEX is used to allow the user to perform more than one maneuver with a given submarine and then shift submarines to perform more maneuvers. If INDEX is less than or equal to zero then the same submarine coefficients will be used for each set of initial conditions. If INDEX is greater than zero the program will read new coefficients as well as a new set of initial conditions.

CHAPTER FOUR - TEST RESULTS AND CONCLUSIONS

IV.1 FUNDAMENTAL MOTION TEST

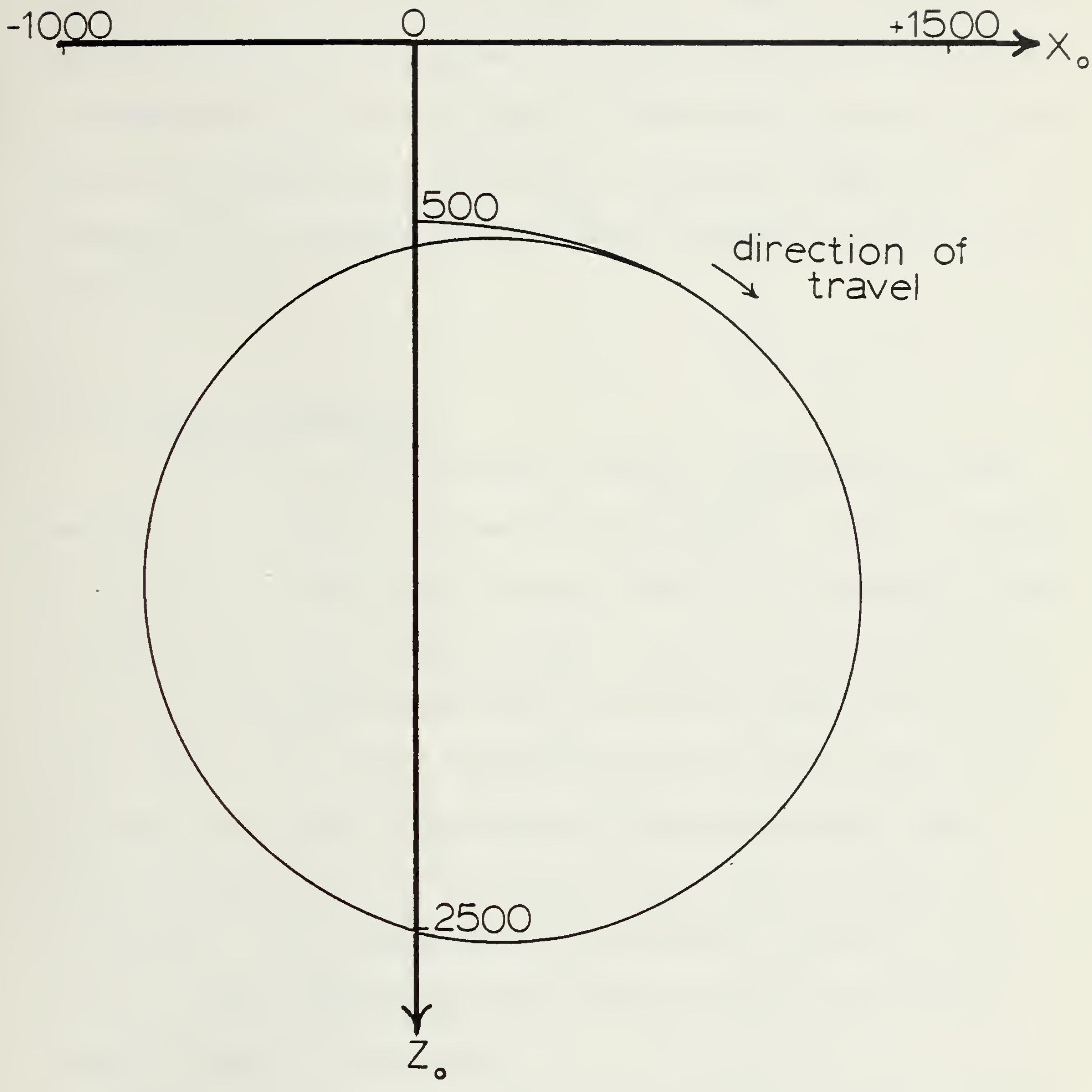
The validation of the simulation model is a matter of importance. The method used was an independent test of each component of the model, and then a series of tests with the components of the model working together. The test maneuvers were selected either because the correct dynamic response was known or because the maneuver was simple enough so that the general nature of the response could be predicted.

The first portion of the model to be tested was the subroutine FUNC. FUNC obtains the solution to the six equations of motion. It was important to establish that these equations were properly installed in the program. A test of subroutine FUNC necessitated a simultaneous test of the MAIN program to read in the hydrodynamic coefficients, set up the "A" matrix for FUNC, perform the Taylor series expansions, and write out the results. The test for FUNC was to reduce the GM of the vessel to zero and then to assume a constant angle on the dive planes. If the model was working correctly it should traverse a perfect circle in the vertical plane.

It was known that the rudder should not move and the model should not roll or yaw. The model should assume a circular trajectory with a constant angular velocity and a constant vertical component of velocity, w. Figure 3 shows the results of the maneuver. A circular trajectory was quickly achieved. The angular velocity was constant, w was constant, and the position of the model in the fixed coordinate system confirmed the circular path. The model did not roll or yaw and the rudder did not move.

IV.2 HORIZONTAL MOTION TEST

With the knowledge that the equations of motion, the Taylor series expansions, and the coordinate system transforms are operating properly, the subroutine RUDDER was the next component to test. It was decided that a simple left turn of 40 degrees would be an appropriate test. A left turn was chosen because that presents the greatest opportunity for error. The original course would be 000 degrees true and the new course would be 320 degrees true. Since the yaw angle is positive when turning right, the model must cope with a negative yaw angle as well as a proper method for dealing with course headings given in true bearing. The subroutines DEPTH and STERN were rendered inoperative to give the opportunity of seeing RUDDER operate without interference. The model would be checked to ensure that the proper rudder rate was used and that the rudder angle did not exceed the maximum angle ordered. The model would be expected to roll and squat in the turn. A change in depth was expected since the dive planes could not act. As the model approached the new heading the



Simulation Model Trajectory With Constant Dive Plane Angle

Figure 3

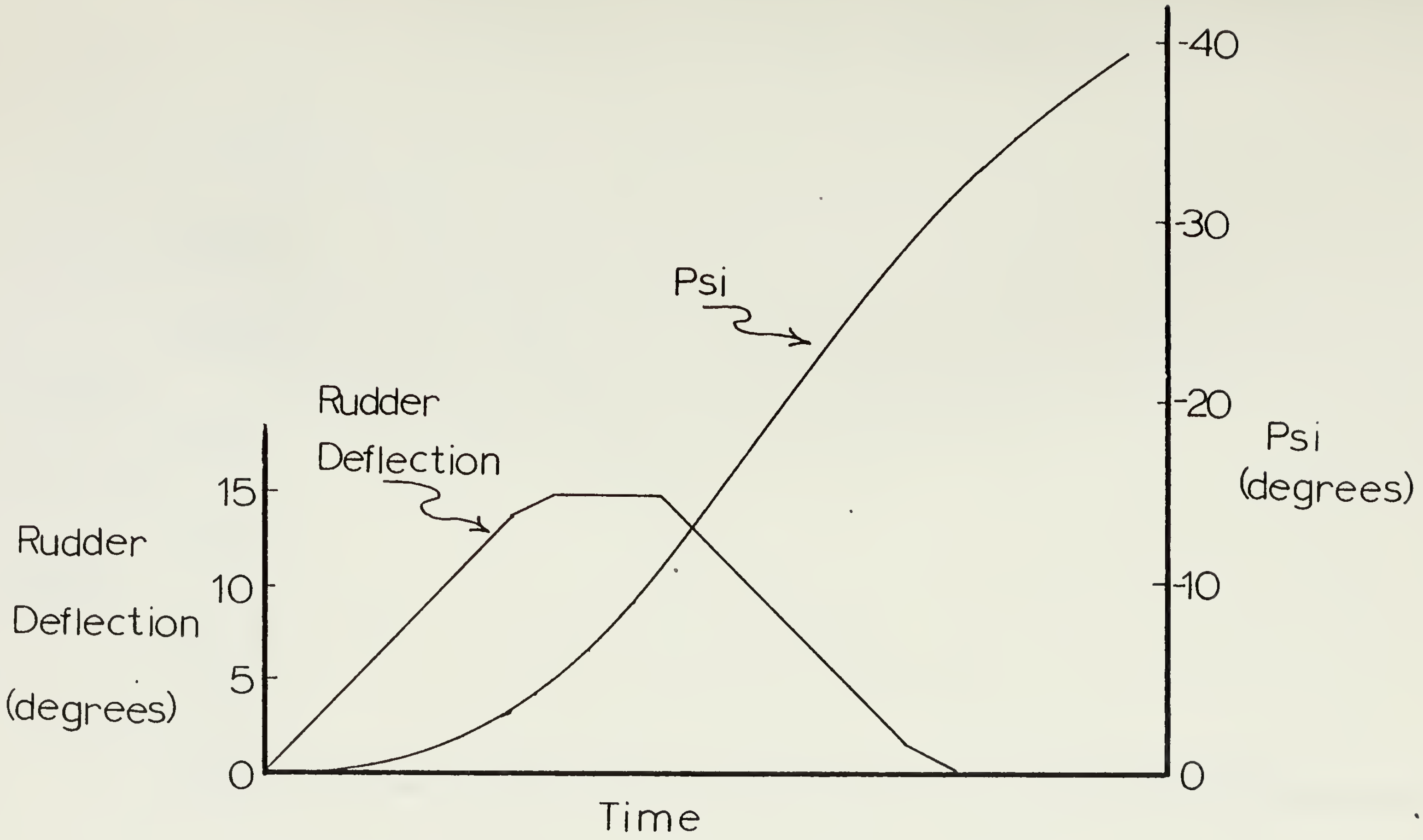
rudder should be put amidships and the vessel should steady out.

In Figure 4 the relationship between the rudder and the heading is shown as a function of time. The model reacted as predicted except possibly at the end of the turn. The rudder was amidships and the new heading was achieved but the program did not run long enough to ensure that the model was steady on the course. The model did roll, squat, and change depth as predicted. The test for RUDDER was considered to be accurate and sufficiently complete to warrant moving to the next test.

IV.3 VERTICAL MOTION TEST

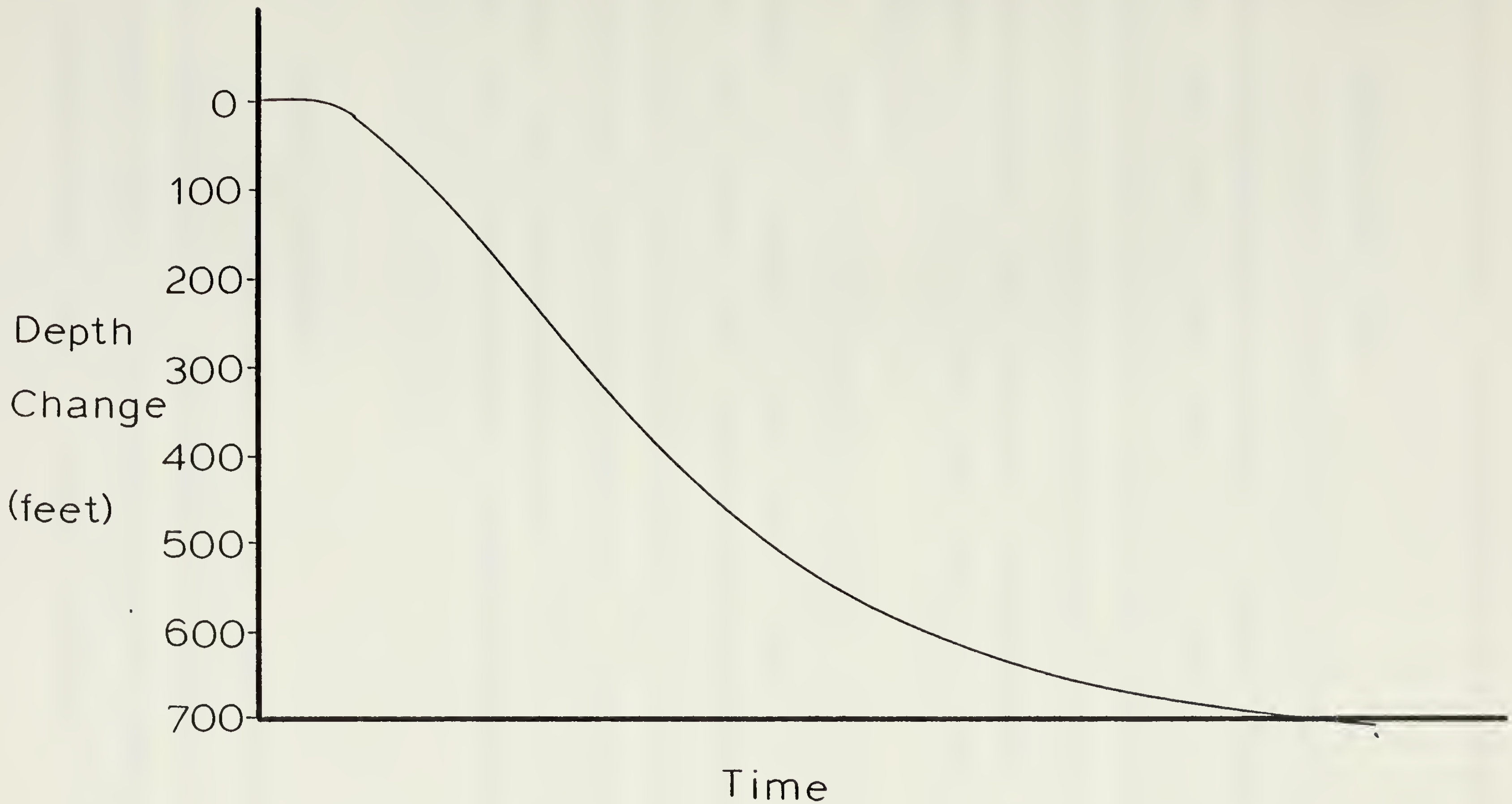
Since DEPTH and STERN operate together to regulate the depth and pitch of the model, they were tested together. The test consisted of a simple dive with a depth change of 700 feet. The course was not changed so the rudder should not move. The model should not roll or yaw. The dive planes should move in the proper direction and at the ordered rate. They should not deflect more than the ordered angle. The model should pitch downward and if the maximum ordered pitch angle is exceeded then the dive planes should move to reduce the pitch angle. As it approaches the desired depth, the model should slow its rate of descent and settle within ten feet of the depth, with 0 ± 1 degree of pitch angle.

The trajectory for this test is shown in Figure 5. The model achieved steady state only six feet beyond the desired depth; the pitch angle was -.03 degrees. The dive planes were all at zero deflection. The model stayed within ten feet of the ordered depth



Rudder Deflection And Angle Of Turn As A Function Of Time

Figure 4



Depth As A Function Of Time For A Simple Dive

Figure 5

and within one degree of zero pitch angle for twenty seconds. This was done to ensure that a steady state had been achieved.

IV.4 COMPLETE MODEL TEST

As a final test the entire simulation model must work together. This test repeated the forty degree left hand turn but this time the dive planes were allowed to react to try to maintain the depth. The program was kept running until the pitch angle was within one degree of being zero, the model was within one degree of the proper course and the depth was within ten feet of the ordered depth. The model was initialized on course 000 degrees true at an initial speed of twenty knots.

The maneuver was successfully completed. The model remained steady at 320 ± 1 degrees true and the final depth was within one foot of the ordered depth. Both the pitch and roll angles were near one degree and were decreasing in magnitude. All of the control surfaces were at zero angle of deflection. It should be noted that the new course had been achieved during the first sixty seconds and that the course was maintained by the model while the proper depth and attitude were being obtained.

IV.5 USE AS A DESIGN TOOL

There are numerous design tasks that could be used to demonstrate the simulation model; a simple example is sufficient for this demonstration. Suppose a designer wanted to know the dynamic effects of increasing the rudder rate in a turn. The designer would elect to

use a simulation model. Since he would not want the dive planes to interfere with the analysis, he would set NODIFF, DCRIT, and NOPICH at large values so that the dive planes would not move. Then he would set RRATE, the rudder rate, at the value he désired to test and order the model to come to a new course. For this run, let's assume that the designer performed the left hand turn from 000 degrees true to 320 degrees true. He used a rudder rate of two degrees/second. He then caused the model to perform the same manuever again with a new rudder rate of four degrees/second. With the output from these two maneuvers he could easily find the time required for the turn, the advance and transfer of the submarine, and the roll and pitch angles as a function of time. The designer could use the information for whatever analysis he had in mind. The model could be run again at new rudder rates or the dive planes could be brought into play or virtually any other maneuver could be simulated. This model can also do snap roll analysis [4].

The cost of running the simulation model is so low that it can be used on a daily basis if desired. For the test turns mentioned above, the cost was less than five dollars, which included reading the cards, compilation, execution, and 900 lines of output. A designer who used the model regularly could have the model as an on line dataset which would greatly reduce the cost of using the model.

IV.6 CONCLUSIONS AND RECOMMENDATIONS

This simulation model does accurately simulate the six degree of freedom motion of a submarine for moderate maneuvers. The cost of

operating the model is very low, which will allow frequent usage. The model is designed to permit a great deal of flexibility in the application of the model. For those designers who use the simulation model, it should be a useful design tool.

If work were continued on this simulation model, it is recommended that some time be spent in improving the appendage control subroutines. Further application of control theory would be helpful with the appendage subroutines. The data input could be organized more efficiently to remove some of the opportunity for error. It could be very useful to have a plotting routine in the model to visually display the information.

REFERENCES

(1) Gertler, M. and Hagen, G. R., "Standard Equations of Motion for Submarine Simulation," NSRDC Report No. 2510, Washington, D.C., June 1967.

(2) Abkowitz, M.A., Stability and Motion Control of Ocean Vehicles, Cambridge, Ma.: The M.I.T. Press, May 1969.

(3) (FORTRAN IV) IBM System/370-360, IMSL Library 1, Edition 6, 1977, Xerox Sigma s/6-7-9-11-560.

(4) Liberatore, D.J., "An Investigation Of Snap Roll in Submarines", M.I.T. Thesis, Cambridge, Ma., May 1977.

(5) ________, IBM System/360 and System/370 FORTRAN IV Language, GC28-6515-10, May 1974.

(6) Comstock, John P., ed. Principles of Naval Architecture, New York: The Society of Naval Architects and Marine Engineers, 1967.

APPENDIX A

A.1 NOTATION

Symbol	Dimensionless Form	Definition
B	$B' = \frac{B}{\frac{1}{2}\rho \ell^2 U^2}$	Buoyancy force, positive upward
CB		Center of buoyancy of submarine
CG		Center of mass of submarine
I_x	$I_x' = \frac{I_x}{\frac{1}{2}\rho \ell^5}$	Moment of inertia of submarine about x axis
I_y	$I_y' = \frac{I_y}{\frac{1}{2}\rho \ell^5}$	Moment of inertia of submarine about y axis
I_z	$I_z' = \frac{I_z}{\frac{1}{2}\rho \ell^5}$	Moment of inertia of submarine about z axis
I_{xy}	$I_{xy}' = \frac{I_{xy}}{\frac{1}{2}\rho \ell^5}$	Product of inertia about xy axis
I_{yz}	$I_{yz}' = \frac{I_{yz}}{\frac{1}{2}\rho \ell^5}$	Product of inertia about yz axes
I_{zx}	$I_{zx}' = \frac{I_{zx}}{\frac{1}{2}\rho \ell^5}$	Product of inertia about zx axes
K	$K' = \frac{K}{\frac{1}{2}\rho \ell^3 U^2}$	Hydrodynamic moment component about x axis (rolling moment)
K_*	$K_*' = \frac{K_*}{\frac{1}{2}\rho \ell^3 U^2}$	Rolling moment when body angle (α, β) and control surface angles are zero
$K_{*\eta}$	$K_{*\eta}' = \frac{K_{*\eta}}{\frac{1}{2}\rho \ell^3 U^2}$	Coefficient used in representing K_* as a function of $(\eta-1)$
K_p	$K_p' = \frac{K_p}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing K as a function of p
$K_{\dot{p}}$	$K_{\dot{p}}' = \frac{K_{\dot{p}}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of $\dot{p}$
$K_{p\|p\|}$	$K_{p\|p\|}' = \frac{K_{p\|p\|}}{\frac{1}{2}\rho \ell^5}$	Second order coefficient used in representing K as a function of p
K_{pq}	$K_{pq}' = \frac{K_{pq}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of the product pq

K_{qr}	$K_{qr}' = \frac{K_{qr}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of the product qr
K_r	$K_r' = \frac{K_r}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing K as a function of r
$K_{\dot{r}}$	$K_{\dot{r}}' = \frac{K_{\dot{r}}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing K as a function of $\dot{r}$
K_v	$K_v' = \frac{K_v}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing K as a function of v
$K_{\dot{v}}$	$K_{\dot{v}}' = \frac{K_{\dot{v}}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of $\dot{v}$
$K_{v\|v\|}$	$K_{v\|v\|}' = \frac{K_{v\|v\|}}{\frac{1}{2}\rho \ell^3}$	Second order coefficient used in representing K as a function of v
K_{vq}	$K_{vq}' = \frac{K_{vq}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of the product vq
K_{vw}	$K_{vw}' = \frac{K_{vw}}{\frac{1}{2}\rho \ell^3}$	Coefficient used in representing K as a function of the product vw
K_{wp}	$K_{wp}' = \frac{K_{wp}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of the product wp
K_{wr}	$K_{wr}' = \frac{K_{wr}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing K as a function of the product wr
$K_{\delta r}$	$K_{\delta r}' = \frac{K_{\delta r}}{\frac{1}{2}\rho \ell^3 U^2}$	First order coefficient used in representing K as a function of δ_r
ℓ	$\ell' = 1$	Overall length of submarine
m	$m' = \frac{m}{\frac{1}{2}\rho \ell^3}$	Mass of submarine, including water in free-flooding spaces
M	$M' = \frac{M}{\frac{1}{2}\rho \ell^3 U^2}$	Hydrodynamic moment component about y axis (pitching moment)
M_*	$M_*' = \frac{M_*}{\frac{1}{2}\rho \ell^3 U^2}$	Pitching moment when body angles (α, β) and control surface angles are zero
M_{pp}	$M_{pp}' = \frac{M_{pp}}{\frac{1}{2}\rho \ell^5}$	Second order coefficient used in representing M as a function of p. First order coefficient is zero.
M_q	$M_q' = \frac{M_q}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing M as a function of q
$M_{q\eta}$	$M_{q\eta}' = \frac{M_{q\eta}}{\frac{1}{2}\rho \ell^4 U}$	First order coefficient used in representing M_q as a function of $(\eta - 1)$
$M_{\dot{q}}$	$M_{\dot{q}}' = \frac{M_{\dot{q}}}{\frac{1}{2}\rho \ell^5}$	Coefficient used in representing M as a function of $\dot{q}$

$M_{q\|q\|}$	$M_{q\|q\|}' = \frac{M_{q\|q\|}}{\frac{1}{2}\rho\ell^5}$	Second order coefficient used in representing M as a function of q
$M_{\|q\|\delta s}$	$M_{\|q\|\delta s}' = \frac{M_{\|q\|\delta s}}{\frac{1}{2}\rho\ell^4 U}$	Coefficient used in representing $M_{\delta s}$ as a function q
M_{rp}	$M_{rp}' = \frac{M_{rp}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing M as a function of the product rp
M_{rr}	$M_{rr}' = \frac{M_{rr}}{\frac{1}{2}\rho\ell^5}$	Second order coefficient used in representing M as a function of r. First order coefficient is zero
M_{vp}	$M_{vp}' = \frac{M_{vp}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing M as a function of the product vp
M_{vr}	$M_{vr}' = \frac{M_{vr}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing M as a function of the product vr
M_{vv}	$M_{vv}' = \frac{M_{vv}}{\frac{1}{2}\rho\ell^3}$	Second order coefficient used in representing M as a function of v
M_w	$M_w' = \frac{M_w}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing M as a function of w
$M_{w\eta}$	$M_{w\eta}' = \frac{M_{w\eta}}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing M_w as a function of $(\eta-1)$
$M_{\dot{w}}$	$M_{\dot{w}}' = \frac{M_{\dot{w}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing M as a function of $\dot{w}$
$M_{\|w\|}$	$M_{\|w\|}' = \frac{M_{\|w\|}}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing M as a function of w; equal to zero for symmetrical function
$M_{\|w\|q}$	$M_{\|w\|q}' = \frac{M_{\|w\|q}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing M_q as a function of w
$M_{w\|w\|}$	$M_{w\|w\|}' = \frac{M_{w\|w\|}}{\frac{1}{2}\rho\ell^3}$	Second order coefficient used in representing M as a function of w
$M_{w\|w\|\eta}$	$M_{w\|w\|\eta}' = \frac{M_{w\|w\|\eta}}{\frac{1}{2}\rho\ell^3}$	First order coefficient used in representing $M_{w\|w\|}$ as a function of $(\eta-1)$
M_{ww}	$M_{ww}' = \frac{M_{ww}}{\frac{1}{2}\rho\ell^3}$	Second order coefficient used in representing M as a function of w; equal to zero for symmetrical function
$M_{\delta b}$	$M_{\delta b}' = \frac{M_{\delta b}}{\frac{1}{2}\rho\ell^3 U^2}$	First order coefficient used in representing M as a function of δ_b
$M_{\delta s}$	$M_{\delta s}' = \frac{M_{\delta s}}{\frac{1}{2}\rho\ell^3 U^2}$	First order coefficient used in representing M as a function of δ_s
$M_{\delta s\eta}$	$M_{\delta s\eta}' = \frac{M_{\delta s\eta}}{\frac{1}{2}\rho\ell^3 U^2}$	First order coefficient used in representing $M_{\delta s}$ as a function of $(\eta-1)$

N	$N' = \frac{N}{\frac{1}{2}\rho\ell^3 U^2}$	Hydrodynamic moment component about z axis (yawing moment)
N_*	$N_*' = \frac{N_*}{\frac{1}{2}\rho\ell^3 U^2}$	Yawing moment when body angles (α, β) and control surface angles are zero
N_p	$N_p' = \frac{N_p}{\frac{1}{2}\rho\ell^4 U}$	First order coefficient used in representing N as a function of p
$N_{\dot{p}}$	$N_{\dot{p}}' = \frac{N_{\dot{p}}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of $\dot{p}$
N_{pq}	$N_{pq}' = \frac{N_{pq}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of the product pq
N_{qr}	$N_{qr}' = \frac{N_{qr}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of the product qr
N_r	$N_r' = \frac{N_r}{\frac{1}{2}\rho\ell^4 U}$	First order coefficient used in representing N as a function of r
$N_{r\eta}$	$N_{r\eta}' = \frac{N_{r\eta}}{\frac{1}{2}\rho\ell^4 U}$	First order coefficient used in representing N_r as a function of $(\eta-1)$
$N_{\dot{r}}$	$N_{\dot{r}}' = \frac{N_{\dot{r}}}{\frac{1}{2}\rho\ell^5}$	Coefficient used in representing N as a function of $\dot{r}$
$N_{r\|r\|}$	$N_{r\|r\|}' = \frac{-N_{r\|r\|}}{\frac{1}{2}\rho\ell^5}$	Second order coefficient used in representing N as a function of r
$N_{\|r\|\delta r}$	$N_{\|r\|\delta r}' = \frac{N_{\|r\|\delta r}}{\frac{1}{2}\rho\ell^4 U}$	Coefficient used in representing $N_{\delta r}$ as a function of r
N_v	$N_v' = \frac{N_v}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing N as a function of v
$N_{v\eta}$	$N_{v\eta}' = \frac{N_{v\eta}}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing N_v as a function of $(\eta-1)$
$N_{\dot{v}}$	$N_{\dot{v}}' = \frac{N_{\dot{v}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N as a function of $\dot{v}$
N_{vq}	$N_{vq}' = \frac{N_{vq}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N as a function of the product vq
$N_{\|v\|r}$	$N_{\|v\|r}' = \frac{N_{\|v\|r}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N_r as a function of v
$N_{v\|v\|}$	$N_{v\|v\|}' = \frac{N_{v\|v\|}}{\frac{1}{2}\rho\ell^3}$	Second order coefficient used in representing N as a function of v
$N_{v\|v\|\eta}$	$N_{v\|v\|\eta}' = \frac{N_{v\|v\|\eta}}{\frac{1}{2}\rho\ell^3}$	First order coefficient used in representing $N_{v\|v\|}$ as a function of $(\eta-1)$

N_{vw}	$N_{vw}' = \frac{N_{vw}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing N as a function of the product vw
N_{wp}	$N_{wp}' = \frac{N_{wp}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N as a function of the product wp
N_{wr}	$N_{wr}' = \frac{N_{wr}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing N as a function of the product wr
$N_{\delta r}$	$N_{\delta r}' = \frac{N_{\delta r}}{\frac{1}{2}\rho\ell^3 U^2}$	First order coefficient used in representing N as a function of δ_r
$N_{\delta r\eta}$	$N_{\delta r\eta}' = \frac{N_{\delta r\eta}}{\frac{1}{2}\rho\ell^3 U^2}$	First order coefficient used in representing $N_{\delta r}$ as a function of $(\eta - 1)$
p	$p' = \frac{p\ell}{U}$	Angular velocity component about x axis relative to fluid (roll)
$\dot{p}$	$\dot{p}' = \frac{\dot{p}\ell^2}{U^2}$	Angular acceleration component about x axis relative to fluid
q	$q' = \frac{q\ell}{U}$	Angular velocity component about y axis relative to fluid (pitch)
$\dot{q}$	$\dot{q}' = \frac{\dot{q}\ell^2}{U^2}$	Angular acceleration component about y axis relative to fluid
r	$r' = \frac{r\ell}{U}$	Angular velocity component about z axis relative to fluid (yaw)
$\dot{r}$	$\dot{r}' = \frac{\dot{r}\ell^2}{U^2}$	Angular acceleration component about z axis relative to fluid
U	$U' = \frac{U}{U}$	Linear velocity of origin of body axes relative to fluid
u	$u' = \frac{u}{U}$	Component of U in direction of the x axis
$\dot{u}$	$\dot{u}' = \frac{\dot{u}\ell}{U^2}$	Time rate of change of u in direction of the x axis
u_c	$u_c' = \frac{u_c}{U}$	Command speed: steady value of ahead speed component u for a given propeller rpm when body angles (α, β) and control surface angles are zero. Sign changes with propeller reversal
v	$v' = \frac{v}{U}$	Component of U in direction of the y axis
$\dot{v}$	$\dot{v}' = \frac{\dot{v}\ell}{U^2}$	Time rate of change of v in direction of the y axis

w	$w' = \frac{w}{U}$	Component of U in direction of the z axis
$\dot{w}$	$\dot{w}' = \frac{\dot{w}\ell}{U^2}$	Time rate of change of w in direction of the z axis
W	$W' = \frac{W}{\frac{1}{2}\rho\ell^2U^2}$	Weight, including water in free flooding spaces
x	$x' = \frac{x}{\ell}$	Longitudinal body axis; also the coordinate of a point relative to the origin of body axes
x_B	$x_B' = \frac{x_B}{\ell}$	The x coordinate of CB
x_G	$x_G' = \frac{x_G}{\ell}$	The x coordinate of CG
x_o	$x_o' = \frac{x_o}{\ell}$	A coordinate of the displacement of CG relative to the origin of a set of fixed axes
X	$X' = \frac{X}{\frac{1}{2}\rho\ell^2U^2}$	Hydrodynamic force component along x axis (longitudinal, or axial, force)
X_{qq}	$X_{qq}' = \frac{X_{qq}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing X as a function of q. First order coefficient is zero
X_{rp}	$X_{rp}' = \frac{X_{rp}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing X as a function of the product rp
X_{rr}	$X_{rr}' = \frac{X_{rr}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing X as a function of r. First order coefficient is zero
$X_{\dot{u}}$	$X_{\dot{u}}' = \frac{X_{\dot{u}}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing X as a function of $\dot{u}$
X_{uu}	$X_{uu}' = \frac{X_{uu}}{\frac{1}{2}\rho\ell^2}$	Second order coefficient used in representing X as a function of u in the non-propelled case. First order coefficient is zero
X_{vr}	$X_{vr}' = \frac{X_{vr}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing X as a function of the product vr
X_{vv}	$X_{vv}' = \frac{X_{vv}}{\frac{1}{2}\rho\ell^2}$	Second order coefiient used in representing X as a function of v. First order coefficient is zero
$X_{vv\eta}$	$X_{vv\eta}' = \frac{X_{vv\eta}}{\frac{1}{2}\rho\ell^2}$	First order coefficient used in representing X_{vv} as a function of $(\eta - 1)$
X_{wq}	$X_{wq}' = \frac{X_{wq}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing X as a function of the product wq

X_{ww}	$X_{ww}' = \frac{X_{ww}}{\frac{1}{2}\rho \ell^2}$	Second order coefficient used in representing X as a function of w. First order coefficient is zero
$X_{ww\eta}$	$X_{ww\eta}' = \frac{X_{ww\eta}}{\frac{1}{2}\rho \ell^2}$	First order coefficient used in representing X_{ww} as a function of $(\eta - 1)$
$X_{\delta b \delta b}$	$X_{\delta b \delta b}' = \frac{X_{\delta b \delta b}}{\frac{1}{2}\rho \ell^2 U^2}$	Second order coefficient used in representing X as a function of δ_b. First order coefficient is zero
$X_{\delta r \delta r}$	$X_{\delta r \delta r}' = \frac{X_{\delta r \delta r}}{\frac{1}{2}\rho \ell^2 U^2}$	Second order coefficient used in representing X as a function of δ_r. First order coefficient is zero
$X_{\delta r \delta r \eta}$	$X_{\delta r \delta r \eta}' = \frac{X_{\delta r \delta r \eta}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing $X_{\delta r \delta r}$ as a function of $(\eta - 1)$
$X_{\delta s \delta s}$	$X_{\delta s \delta s}' = \frac{X_{\delta s \delta s}}{\frac{1}{2}\rho \ell^2 U^2}$	Second order coefficient used in representing X as a function of δ_s. First order coefficient is zero
$X_{\delta s \delta s \eta}$	$X_{\delta s \delta s \eta}' = \frac{X_{\delta s \delta s \eta}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing $X_{\delta s \delta s}$ as a function of $(\eta - 1)$
y	$y' = \frac{y}{\ell}$	Lateral body axis; also the coordinate of a point relative to the origin of body axes
y_B	$y_B' = \frac{y_B}{\ell}$	The y coordinate of CB
y_G	$y_G' = \frac{y_G}{\ell}$	The y coordinate of CG
y_o	$y_o' = \frac{y_o}{\ell}$	A coordinate of the displacement of CG relative to the origin of a set of fixed axes
Y	$Y' = \frac{Y}{\frac{1}{2}\rho \ell^2 U^2}$	Hydrodynamic force component along y axis (lateral force)
Y_*	$Y_*' = \frac{Y}{\frac{1}{2}\rho \ell^2 U^2}$	Lateral force when body angles (α, β) and control surface angles are zero
Y_p	$Y_p' = \frac{Y_p}{\frac{1}{2}\rho \ell^3 U}$	First order coefficient used in representing Y as a function of p
$Y_{\dot{p}}$	$Y_{\dot{p}}' = \frac{Y_{\dot{p}}}{\frac{1}{2}\rho \ell^4}$	Coefficient used in representing Y as a function of $\dot{p}$
$Y_{p\|p\|}$	$Y_{p\|p\|}' = \frac{Y_{p\|p\|}}{\frac{1}{2}\rho \ell^4}$	Second order coefficient used in representing Y as a function of p

Y_{pq}	$Y_{pq}' = \frac{Y_{pq}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Y as a function of the product pq
Y_{qr}	$Y_{qr}' = \frac{Y_{qr}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Y as a function of the product qr
Y_r	$Y_r' = \frac{Y_r}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing Y as a function of r
$Y_{r\eta}$	$Y_{r\eta}' = \frac{Y_{r\eta}}{\frac{1}{2}\rho\ell^3 U}$	First order coefficient used in representing Y_r as a function of $(\eta - 1)$
$Y_{\dot{r}}$	$Y_{\dot{r}}' = \frac{Y_{\dot{r}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Y as a function of $\dot{r}$
$Y_{\|r\|\delta r}$	$Y_{\|r\|\delta r}' = \frac{Y_{\|r\|\delta r}}{\frac{1}{2}\rho\ell^3 U}$	Coefficient used in representing $Y_{\delta r}$ as a function of r
Y_v	$Y_v' = \frac{Y_v}{\frac{1}{2}\rho\ell^2 U}$	First order coefficient used in representing Y as a function of v
$Y_{v\eta}$	$Y_{v\eta}' = \frac{Y_{v\eta}}{\frac{1}{2}\rho\ell^2 U}$	First order coefficient used in representing Y_v as a function of $(\eta - 1)$
$Y_{\dot{v}}$	$Y_{\dot{v}}' = \frac{Y_{\dot{v}}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Y as a function of $\dot{v}$
Y_{vq}	$Y_{vq}' = \frac{Y_{vq}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Y as a function of the product vq
$Y_{v\|r\|}$	$Y_{v\|r\|}' = \frac{Y_{v\|r\|}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Y_v as a function of r
$Y_{v\|v\|}$	$Y_{v\|v\|}' = \frac{Y_{v\|v\|}}{\frac{1}{2}\rho\ell^2}$	Second order coefficient used in representing Y as a function of v
$Y_{v\|v\|\eta}$	$Y_{v\|v\|\eta}' = \frac{Y_{v\|v\|\eta}}{\frac{1}{2}\rho\ell^2}$	First order coefficient used in representing $Y_{v\|v\|}$ as a function of $(\eta - 1)$
Y_{vw}	$Y_{vw}' = \frac{Y_{vw}}{\frac{1}{2}\rho\ell^2}$	Coefficient used in representing Y as a function of the product vw
Y_{wp}	$Y_{wp}' = \frac{Y_{wp}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Y as a function of the product wp
Y_{wr}	$Y_{wr}' = \frac{Y_{wr}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Y as a function of the product wr
$Y_{\delta r}$	$Y_{\delta r}' = \frac{Y_{\delta r}}{\frac{1}{2}\rho\ell^2 U^2}$	First order coefficient used in representing Y as a function of δr
$Y_{\delta r\eta}$	$Y_{\delta r\eta}' = \frac{Y_{\delta r\eta}}{\frac{1}{2}\rho\ell^2 U^2}$	First order coefficient used in representing $Y_{\delta r}$ as a function of $(\eta - 1)$

z	$z' = \frac{z}{\ell}$	Normal body axis; also the coordinate of a point relative to the origin of body axes
z_B	$z_B' = \frac{z_B}{\ell}$	The z coordinate of CB
z_G	$z_G' = \frac{z_G}{\ell}$	The z coordinate of CG
z_o	$z_o' = \frac{z_o}{\ell}$	A coordinate of the displacement of CG relative to the origin of a set of fixed axes
Z	$Z' = \frac{Z}{\frac{1}{2}\rho\ell^2U^2}$	Hydrodynamic force component along z axis (normal force)
Z_*	$Z_*' = \frac{Z_*}{\frac{1}{2}\rho\ell^2U^2}$	Normal force when body angles (α, β) and control surface angles are zero
Z_{pp}	$Z_{pp}' = \frac{Z_{pp}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing Z as a function of p. First order coefficient is zero
Z_q	$Z_q' = \frac{Z_q}{\frac{1}{2}\rho\ell^3U}$	First order coefficient used in representing Z as a function of q
$Z_{q\eta}$	$Z_{q\eta}' = \frac{Z_{q\eta}}{\frac{1}{2}\rho\ell^3U}$	First order coefficient used in representing Z_q as a function of (η-1)
$Z_{\dot{q}}$	$Z_{\dot{q}}' = \frac{Z_{\dot{q}}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Z as a function of $\dot{q}$
$Z_{\lvert q \rvert \delta s}$	$Z_{\lvert q \rvert \delta s}' = \frac{Z_{\lvert q \rvert \delta s}}{\frac{1}{2}\rho\ell^3U}$	Coefficient used in representing $Z_{\delta s}$ as a function of q
Z_{rp}	$Z_{rp}' = \frac{Z_{rp}}{\frac{1}{2}\rho\ell^4}$	Coefficient used in representing Z as a function of the product rp
Z_{rr}	$Z_{rr}' = \frac{Z_{rr}}{\frac{1}{2}\rho\ell^4}$	Second order coefficient used in representing Z as a function of r. First order coefficient is zero
Z_w	$Z_w' = \frac{Z_w}{\frac{1}{2}\rho\ell^2U}$	First order coefficient used in representing Z as a function of w
$Z_{w\eta}$	$Z_{w\eta}' = \frac{Z_{w\eta}}{\frac{1}{2}\rho\ell^2U}$	First order coefficient used in representing Z_w as a function of (η-1)
$Z_{\dot{w}}$	$Z_{\dot{w}}' = \frac{Z_{\dot{w}}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Z as a function of $\dot{w}$
$Z_{\lvert w \rvert}$	$Z_{\lvert w \rvert}' = \frac{Z_{\lvert w \rvert}}{\frac{1}{2}\rho\ell^2U}$	First order coefficient used in representing Z as a function of w; equal to zero for symmetrical function
$Z_{w\lvert q \rvert}$	$Z_{w\lvert q \rvert}' = \frac{Z_{w\lvert q \rvert}}{\frac{1}{2}\rho\ell^3}$	Coefficient used in representing Z_w as a function of q

$Z_{w\|w\|}$	$Z_{w\|w\|}' = \frac{Z_{w\|w\|}}{\frac{1}{2}\rho \ell^2}$	Second order coefficient used in representing Z as a function of w
$Z_{w\|w\|\eta}$	$Z_{w\|w\|\eta}' = \frac{Z_{w\|w\|\eta}}{\frac{1}{2}\rho \ell^2}$	First order coefficient used in representing $Z_{w\|w\|}$ as a function of $(\eta - 1)$
Z_{ww}	$Z_{ww}' = \frac{Z_{ww}}{\frac{1}{2}\rho \ell^2}$	Second order coefficient used in representing Z as a function of w; equal to zero for symmetrical function
$Z_{\delta b}$	$Z_{\delta b}' = \frac{Z_{\delta b}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing Z as a function of δ_b
$Z_{\delta s}$	$Z_{\delta s}' = \frac{Z_{\delta s}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing Z as a function of δ_s
$Z_{\delta s \eta}$	$Z_{\delta s \eta}' = \frac{Z_{\delta s \eta}}{\frac{1}{2}\rho \ell^2 U^2}$	First order coefficient used in representing $Z_{\delta s}$ as a function of $(\eta - 1)$
α		Angle of attack
β		Angle of drift
δ_b		Deflection of bowplane or sailplane
δ_r		Deflection of rudder
δ_s		Deflection of sternplane
η		The ratio $\frac{u_c}{U}$
θ		Angle of pitch
ψ		Angle of yaw
ϕ		Angle of roll
a_i, b_i, c_i		Sets of constants used in the representation of propeller thrust in the axial equation

A.2 EQUATIONS OF MOTION

AXIAL FORCE

$$m\left[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})\right] =$$

$$+ \frac{\rho}{2}\ell^4\left[X_{qq}'\,q^2 + X_{rr}'\,r^2 + X_{rp}'\,rp\right]$$

$$+ \frac{\rho}{2}\ell^3\left[X_{\dot{u}}'\,\dot{u} + X_{vr}'\,vr + X_{wq}'\,wq\right]$$

$$+ \frac{\rho}{2}\ell^2\left[X_{uu}'\,u^2 + X_{vv}'\,v^2 + X_{ww}'\,w^2\right]$$

$$+ \frac{\rho}{2}\ell^2u^2\left[X_{\delta r\,\delta r}'\,\delta r^2 + X_{\delta s\,\delta s}'\,\delta s^2 + X_{\delta b\,\delta b}'\,\delta b^2\right]$$

$$+ \tfrac{1}{2}\rho\ell^2\left[a_i\,u^2 + b_i\,uu_c + c_i\,u_c^{\,2}\right]$$

$$- (W - B)\sin\theta$$

$$+ \frac{\rho}{2}\ell^2\left[X_{vv\eta}'\,v^2 + X_{ww\eta}'\,w^2 + X_{\delta r\delta r\eta}'\,\delta_r^{\,2}\,u^2\right.$$

$$\left. + X_{\delta s\delta s\eta}\,\delta_s^{\,2}\,u^2\right](\eta - 1)$$

LATERAL FORCE

$$m\left[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})\right] =$$

$$+\frac{\rho}{2}\ell^4\left[Y_{\dot{r}}'\dot{r} + Y_{\dot{p}}'\dot{p} + Y_{p|p|}'p|p| + Y_{pq}'pq + Y_{qr}'qr\right]$$

$$+\frac{\rho}{2}\ell^3\left[Y_{\dot{v}}'\dot{v} + Y_{vq}'vq + Y_{wp}'wp + Y_{wr}'wr\right]$$

$$+\frac{\rho}{2}\ell^3\left[Y_r'ur + Y_p'up + Y_{|r|\delta r}'u|r|\delta r + Y_{v|r|}'\frac{v}{|v|}\left|(v^2 + w^2)^{\frac{1}{2}}\right||r|\right]$$

$$+\frac{\rho}{2}\ell^2\left[Y_*'u^2 + Y_v'uv + Y_{v|v|}'v\left|(v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2}\ell^2\left[Y_{vw}'vw + Y_{\delta r}'u^2\delta r\right]$$

$$+(W - B)\cos\theta\sin\phi$$

$$+\frac{\rho}{2}\ell^3 Y_{r\eta}'ur(\eta - 1)$$

$$+\frac{\rho}{2}\ell^2\left[Y_{v\eta}'uv + Y_{v|v|\eta}'v\left|(V^2 + w^2)^{\frac{1}{2}}\right| + Y_{\delta r\eta}'\delta_r u^2\right](\eta - 1)$$

NORMAL FORCE

$$m\left[\dot{w} - uq + vp - z_G (p^2 + q^2) + x_G (rp - \dot{q}) + y_G (rq + \dot{p})\right] =$$

$$+\frac{\rho}{2} \ell^4 \left[Z_{\dot{q}}{}' \dot{q} + Z_{pp}{}' p^2 + Z_{rr}{}' r^2 + Z_{rp}{}' rp\right]$$

$$+\frac{\rho}{2} \ell^3 \left[Z_{\dot{w}}{}' \dot{w} + Z_{vr}{}' vr + Z_{vp}{}' vp\right]$$

$$+\frac{\rho}{2} \ell^3 \left[Z_q{}' uq + Z_{|q|\delta s}{}' u|q|\delta s + Z_{w|q|}{}' \frac{w}{|w|}\left|(v^2 + w^2)^{\frac{1}{2}}\right| |q|\right]$$

$$+\frac{\rho}{2} \ell^2 \left[Z_*{}' u^2 + Z_w{}' uw + Z_{w|w|}{}' w\left|(v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2} \ell^2 \left[Z_{|w|}{}' u|w| + Z_{ww}{}' \left|w (v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2} \ell^2 \left[Z_{vv}{}' v^2 + Z_{\delta s}{}' u^2 \delta s + Z_{\delta b}{}' u^2 \delta b\right]$$

$$+ (W - B) \cos \theta \cos \phi$$

$$+ \frac{\rho}{2} \ell^3 Z_{q\eta}{}' uq (\eta - 1)$$

$$+\frac{\rho}{2} \ell^2 \left[Z_{w\eta}{}' uw + Z_{w|w|\eta}{}' w\left|(v^2 + w^2)^{\frac{1}{2}}\right| + Z_{\delta s\eta}{}' \delta_s u^2\right](\eta - 1)$$

ROLLING MOMENT

$$I_x\dot{p} + (I_z - I_y)\,qr - (\dot{r} + pq)\,I_{xz} + (r^2 - q^2)\,I_{yz} + (pr - \dot{q})\,I_{xy}$$

$$+ m\left[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)\right] =$$

$$+\frac{\rho}{2}\,\ell^5\left[K_{\dot{p}}{}'\,\dot{p} + K_{\dot{r}}{}'\,\dot{r} + K_{qr}{}'\,qr + K_{pq}{}'\,pq + K_{p|p|}{}'\,p|p|\right]$$

$$+\frac{\rho}{2}\,\ell^4\left[K_p{}'\,up + K_r{}'\,ur + K_{\dot{v}}{}'\,\dot{v}\right]$$

$$+\frac{\rho}{2}\,\ell^4\left[K_{vq}{}'\,vq + K_{wp}{}'\,wp + K_{wr}{}'\,wr\right]$$

$$+\frac{\rho}{2}\,\ell^3\left[K_*{}'\,u^2 + K_v{}'\,uv + K_{v|v|}{}'\,v\left|(v^2 + w^2)^{\frac{1}{2}}\right|\right]$$

$$+\frac{\rho}{2}\,\ell^3\left[K_{vw}{}'vw + K_{\delta r}{}'\,u^2\,\delta r\right]$$

$$+ (y_G W - y_B B)\cos\theta\cos\phi - (z_G W - z_B B)\cos\theta\sin\phi\ \Big]$$

$$+\frac{\rho}{2}\,\ell^3\,K_{*\eta}{}'\,u^3\,(\eta - 1)$$

PITCHING MOMENT

$$I_y \dot{q} + (I_x - I_z)\, rp - (\dot{p} + qr)\, I_{xy} + (p^2 - r^2)\, I_{zx} + (qp - \dot{r})\, I_{yz}$$

$$+ m \left[z_G (\dot{u} - vr + wq) - x_G (\dot{w} - uq + vp) \right] =$$

$$+ \frac{\rho}{2} \ell^5 \left[M_{\dot{q}}' \dot{q} + M_{pp}' p^2 + M_{rr}' r^2 + M_{rp}' rp + M_{q|q|}' q|q| \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[M_{\dot{w}}' \dot{w} + M_{vr}' vr + M_{vp}' vp \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[M_q' uq + M_{|q|\delta s}' u|q|\delta s + M_{|w|q}' |(v^2 + w^2)^{\frac{1}{2}}| q \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[M_*' u^2 + M_w' uw + M_{w|w|}' w \left| (v^2 + w^2)^{\frac{1}{2}} \right| \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[M_{|w|}' u|w| + M_{ww}' \left| w (v^2 + w^2)^{\frac{1}{2}} \right| \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[M_{vv}' v^2 + M_{\delta s}' u^2 \delta s + M_{\delta b}' u^2 \delta b \right]$$

$$- (x_G W - x_B B) \cos\theta \cos\phi - (z_G W - z_B B) \sin\theta$$

$$+ \frac{\rho}{2} \ell^4 M_{q\eta}' uq (\eta - 1)$$

$$+ \frac{\rho}{2} \ell^3 \left[M_{w\eta}' uw + M_{w|w|\eta}' w \left| (v^2 + w^2)^{\frac{1}{2}} \right| + M_{\delta s\eta}' \delta_s u^2 \right] (\eta - 1)$$

YAWING MOMENT

$$I_z \dot{r} + (I_y - I_x)\, pq - (\dot{q} + rp)\, I_{yz} + (q^2 - p^2)\, I_{xy} + (rq - \dot{p})\, I_{zx}$$

$$+ m \left[x_G (\dot{v} - wp + ur) - y_G (\dot{u} - vr + wq) \right] =$$

$$+ \frac{\rho}{2} \ell^5 \left[N_{\dot{r}}{}' \dot{r} + N_{\dot{p}}{}' \dot{p} + N_{pq}{}' pq + N_{qr}{}' qr + N_{r|r|}{}' r|r| \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[N_{\dot{v}}{}' \dot{v} + N_{wr}{}' wr + N_{wp}{}' wp + N_{vq}{}' vq \right]$$

$$+ \frac{\rho}{2} \ell^4 \left[N_p{}' up + N_r{}' ur + N_{|r|\delta r}{}' u|r|\delta r + N_{|v|r}{}' \left|(v^2 + w^2)^{\frac{1}{2}}\right| r \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[N_*{}' u^2 + N_v{}' uv + N_{v|v|}{}' v \left|(v^2 + w^2)^{\frac{1}{2}}\right| \right]$$

$$+ \frac{\rho}{2} \ell^3 \left[N_{vw}{}' vw + N_{\delta r}{}' u^2 \delta r \right]$$

$$+ (x_G W - x_B B) \cos\theta \sin\phi + (y_G W - y_B B) \sin\theta$$

$$+ \frac{\rho}{2} \ell^4 N_{r\eta}{}' ur (\eta - 1)$$

$$+ \frac{\rho}{2} \ell^3 \left[N_{v\eta}{}' uv + N_{v|v|\eta}{}' v \left|(v^2 + w^2)^{\frac{1}{2}}\right| + N_{\delta r\eta}{}' \delta_r u^2 \right] (\eta - 1)$$

A.3 AXIS TRANSFORMATIONS [2]

1) A transformation of an axes system takes a quantity described in one frame of reference and transforms it into another frame of reference such that if we measured the same quantity in the second frame of reference the transformed quantity and the measured quantity would be identical.

2) Transforms between frames are needed in the study of the motions of ocean vehicles because the equations of motion for such a vehicle are most easily derived in the inertial frame attached to the earth (x_o, y_o, z_o) frame, while the forces acting on the vehicle are most easily evaluated in the frame attached to the vehicle (x, y, z). Hence, we ultimately desire to transform the equations of motion from the inertial frame into the non-inertial frame fixed in the vehicle.

3) If $\vec{V}_o$ is some vector measure in the x_o, y_o, z_o frame and $\vec{V}$ some vector measured in the x, y, z frame which is only changed in orientation then:

$$\vec{V} = T(\psi, \theta, \phi)\,\vec{V}_o \text{ where } T(\psi, \theta, \phi) = \text{the transform.}$$

Where

$$T(\phi, \theta, \psi) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\sin\psi\cos\phi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}$$

4) If $\vec{V}_o$ and $\vec{V}$ are the same vectors as in (3) above, then:

$$\vec{V}_o = T^{-1}(\phi, \theta, \psi)\,\vec{V}$$

Where

$$T^{-1}(\phi, \theta, \psi) =$$

$$\begin{bmatrix} \cos\theta\cos\psi & -\sin\psi\cos\phi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

APPENDIX B

B.1 LIST OF VARIABLES

VARIABLE	MEANING
A	Six by six matrix containing coefficients of UDT, VDT, WDT, PDT, QDT, RDT.
AA	Six by six matrix set equal to matrix A before each call to FUNC. The values of AA are lost in the matrix reduction performed by LEQT1F.
A1,A2	Limits used for selecting the proper propeller thrust.
AI,BI,CI	Set of constants representing the propeller thrust in the X-equation.
B	Ship's buoyancy.
BORATE	Average bowplane rate.
BOWMAX	Maximum ordered bowplane deflection.
COURSE	New course for the ship.
DECRIT	Value of depth error when dive planes are returned to zero deflection.
DEL1	Calculated deflection of the bowplane.
DEL3	Calculated deflection of the stern plane.

DEL4	Calculated deflection of the rudder.
DELB	Actual bowplane deflection at time t.
DELBO	Actual bowplane deflection (units changed for output).
DELGM	Amount the original GM is to be changed.
DELR	Actual rudder deflection at time t.
DELRO	Actual rudder deflection (units changed for output).
DELS	Actual stern plane deflection at time t.
DELSO	Actual stern plane deflection (units changed for output).
DELT	Time increment used in iteration.
DIFF	Difference between present depth and ordered depth.
E	Six by one matrix containing solution to right hand side of equations of motion.
ICNT	Counter used to count the number of iterations
ID	Forty-character alphanumeric heading.
INDEX	If greater than zero, read new submarine coefficients. If less than or equal to zero, run same submarine for new initial

	conditions.
IX,IY,IZ,IXY,IXZ,IYZ	Moments of inertia.
KCOEFF	K - equation coefficients.
L	Ship's overall length.
LEQT1F	Matrix reduction subroutine from IMSLIB [3]
M	Ship's mass.
MAXANG	Maximum ordered dive/ascent angle.
MCOEFF	M - equation coefficients.
NCOEFF	N - equation coefficients.
NODIFF	Acceptable error range around ordered depth.
NOPICH	Acceptable error range around zero pitch angle.
ODEPTH	Ordered depth.
ONCRS	Acceptable error range around ordered course.
P,Q,R	Angular velocity about the x, y, and z axes respectively.
PDT,QDT,RDT	Angular acceleration about the x, y, and z axes respectively.
PDTO,QDTO,RDTO	Angular acceleration about the x, y, and z

	axes, respectively (with units changed for output).
PHI	Angle of roll.
PHIO	Angle of roll (with units changed for output).
PO,QO,RO	Angular velocity about the x, y, and z axes respectively (with units changed for output).
PSI	Angle of yaw.
PSIO	Angle of yaw (with units changed for output).
RHO	Sea water density.
RRATE	Average rudder rate.
RUDAMT	Maximum ordered rudder deflection.
STERAT	Average stern plane rate.
STERMX	Maximum ordered stern plane deflection.
STOW	Storage array for half of output data.
T	Present time.
T1 . . . T18	Time lag signals.
THETA	Angle of pitch.

THETAO	Angle of pitch (with units changed for output).
TLAGB	Time lag for bowplane control system.
TLAGR	Time lag for the rudder control system.
TLAGS	Time lag for stern plane control system.
U,V,W	Forward, lateral, and vertical velocities respectively.
UDT,VDT,WDT	Forward, lateral, and vertical accelerations respectively (with units changed for output).
UO	Initial forward velocity.
UOO,VO,WO	Forward, lateral, and vertical velocities respectively (with units changed for output).
WT	Ship's weight.
X,Y,Z	Coordinate labels of the fixed (x_o, y_o, z_o) coordinate system.
XB,YB,ZB	The x, y, z position of the center of buoyancy.
XCOEFF	X - equation coefficients
XDT,YDT,ZDT	Velocities in the fixed coordinate system along the x_o, y_o, z_o axes respectively.
XG,YG,ZG	The x, y, z position of the center of

gravity.

YCOEFF Y - equation coefficients

ZCOEFF Z - equation coefficients

B.2 COMPUTER PROGRAM LISTING

```
C
C     MAIN PROGRAM --- READ DATA & INITIAL CONDITIONS & INITIAL VALUES.
C                      INITIALIZE & NON-DIMENSIONALIZE.  CALCULATE LINEAR D.E.
C                      COEFFICIENTS, ACCELERATIONS, VELOCITIES AND WRITE OUTPUT.
C
      IMPLICIT REAL*8(A-H),REAL*8(L-Z)
      REAL*8 XCOEFF(16),YCOEFF(22),ZCOEFF(22),KCOEFF(17),MCOEFF(23),NCOE
     1FF(22),AA(6,6),E(6),AI(3),BI(3),CI(3),WKAREA(6,6)
      REAL*8 K1,K2,K3,K4,K5,K6,K7,K8,K9,K10
      REAL*8 IX,IY,IZ,IXY,IXZ,IYZ
      REAL*8 A(6,6),STOW(600,11)
      REAL ID(40)
      COMMON /ONE/ K3,K4,K5,K6,K7,K8,T3,T4,T7,T8,T9,T10,TLAGS,STERAT,STE
     1RMX
      COMMON /TWO/ K1,K2,T1,T2,T13,T14,T15,T16,TLAGB,BORATE,BOWMAX,T,CHE
     1CK
      COMMON /THREE/ K9,K10,T5,T6,T11,T12,T17,T18,TLAGR,RRATE,RUDAMT,COU
     1RSE
      COMMON /FOUR/ U,V,W,UO,PHI,AI,BI,CI,XCOEFF,YCOEFF,ZCOEFF,KCOEFF,MC
     1OEFF,NCOEFF,L,M,P,RHO,WT,B,XG,YG,ZG,IZ,IX,IY,IXZ,IXY,IYZ,RDT,QDT,P
     2DT,WDT,VDT,UDT,XB,YB,ZB,A1,A2,AA,WKAREA
      COMMON /FIVE/ THETA,Q,DELS,DELB,DELT,PSI,R,DELR
      COMMON /SIX/ DIFF,ADIFF,ZDT,ATHETA,MAXANG,NOPICH,DCRIT
C
C     READ NON-DIMENSIONAL COEFFICIENTS
C
      INDEX=0
    1 READ(5,16,END=999) INDEX
      IF(INDEX) 2,2,3
    3 READ(5,10,END=999) (XCOEFF(I),I=1,16)
      READ(5,10) (YCOEFF(I),I=1,22)
      READ(5,10) (ZCOEFF(I),I=1,22)
      READ(5,10) (KCOEFF(I),I=1,17)
      READ(5,10) (MCOEFF(I),I=1,23)
      READ(5,10) (NCOEFF(I),I=1,22)
```

```
      READ(5,10) (AI(I),I=1,3)
      READ(5,10) (BI(I),I=1,3)
      READ(5,10) (CI(I),I=1,3)
      READ(5,13) A1,A2
C
C     READ SHIP CHARACTERISTICS --- MASS, LENGTH, ETC.
C
      READ(5,12) XG,YG,ZG
      READ(5,12) XB,YB,ZB
      READ(5,12) M,L,RHO
      READ(5,12) IX,IY,IZ
      READ(5,12) IXY,IXZ,IYZ
      READ(5,13) WT,B
C
C     READ CONSTANTS FOR SHIP CONTROL SURFACES
C
      READ(5,13) K1,K2
      READ(5,11) K5,K6,K7,K8
      READ(5,13) K9,K10
C
C     READ INITIAL CONDITIONS --- DEPTH, SPEED, ANGLES, ETC.
C
    2 READ(5,14,END=999)   (ID(I),I=1,40)
      ICNT=1
      READ(5,15) U,V,W,UDT,VDT,WDT
      READ(5,15) P,Q,R,PDT,QDT,RDT
      READ(5,12) THETA,PHI,PSI
      READ(5,18) DELS,DELB,DELR,DELGM,Z
C
C     READ SENSITIVITY PARAMETERS FOR COURSE, DEPTH & PITCH
C
      READ(5,11) NODIFF,NOPICH,ONCRS,DCRIT
C
C     READ COURSE CHANGE INFORMATION
C
      READ(5,18) TLAGR,COURSE,RRATE,RUDAMT,DELT
```

```
C
C      READ DEPTH CHANGE INFORMATION
C
       READ(5,19)BORATE,BOWMAX,STERAT,STERMX,MAXANG,ODEPTH,TLAGB,TLAGS
    10 FORMAT(D10.4)
    11 FORMAT(4D10.4)
    12 FORMAT(3D10.4)
    13 FORMAT(2D10.4)
    14 FORMAT(40A1)
    15 FORMAT(6D10.4)
    16 FORMAT (I2)
    18 FORMAT(5D10.4)
    19 FORMAT (8D10.4)
C
C      WRITE HEADING AND INPUTS
C
       WRITE(6,49)   (ID(I),I=1,40)
    49 FORMAT(1H1,39X,'**************************************************
      1*',/,40X,'*',49X,'*',/,40X,'*    ',40A1,'     *',/,40X,'*',49X,'*'
      2,/,40X,'***************************************************',///)
       WRITE(6,42)U,Z,DELT,ODEPTH,COURSE,RRATE,BORATE,STERAT,RUDAMT,BOWMA
      1X,STERMX,MAXANG
    42 FORMAT(20X,'BASE SPEED =',F3.0,' (KTS)',10X,'STARTING DEPTH =',F5.
      10,' (FT)',9X,'STARTING COURSE = 000 (DEG)',//,20X,'DELTA T =',F3.1
      2,' (SEC)',13X,'ORDERED DEPTH =',F5.0,' (FT)',10X,'ORDERED COURSE =
      3',F4.0,' (DEG)',//,20X,'RUDDER RATE =',F4.1,' (DEG/SEC)',4X,'BOW P
      4LANE RATE =',F4.1,' (DEG/SEC)',5X,'STERN PLANE RATE =',F4.1,' (DEG
      5/SEC)',//,20X,'MAX RUDDER ALLOWED =',F4.1,' (DEG)',17X,'MAX BOW PL
      6ANE ANGLE ALLOWED =',F4.1,' (DEG)',//,20X,'MAX STERN PLANE ANGLE A
      7LLOWED =',F4.1,' (DEG)',6X,'MAX DIVE ANGLE ALLOWED =',F4.1,' (DEG)
      8',/////)
       WRITE(6,51)
    51 FORMAT(5X,'T',11X,'X',10X,'Y',10X,'Z',10X,'U',10X,'V',10X,'W',9X,'
      1UDT',8X,'VDT',8X,'WDT',7X,'DELS',7X,'DELB')
       WRITE(6,52)
    52 FORMAT(3X,'(SEC)',7X,'(FT)',7X,'(FT)',7X,'(FT)',7X,'(KTS)',6X,'(KT
```

```
     1S)',6X,'(KTS)',4X,'(KTS/SEC)',2X,'(KTS/SEC)',2X,'(KTS/SEC)',3X,'(D
     2EG)',7X,'(DEG)')
C
C      INITIALIZE STORAGE MATRIX
C
       DO 46 J=1,11
       DO 46 JJ=1,600
       STOW(JJ,J)=0.0D1
    46 CONTINUE
       J=1
C
C      IF GM IS BEING CHANGED CALCULATE NEW VERTICAL CENTER OF BUOYANCY
C
       ZB=ZB-DELGM
C
C         NON-DIMENSIONALIZE MASS AND MOMENTS OF INERTIA
C
       M=M/(.5*RHO*(L**3))
       DENOM=.5*RHO*(L**5)
       IX=IX/DENOM
       IY=IY/DENOM
       IZ=IZ/DENOM
       IXY=IXY/DENOM
       IYZ=IYZ/DENOM
       IXZ=IXZ/DENOM
C
C         CALCULATE COEFFICIENT VALUES FOR THE SIX LINEAR D.E.'S
C
       A(1,1)=M-XCOEFF(4)
       A(1,2)=0.0
       A(1,3)=0.0
       A(1,4)=0.0
       A(1,5)=M*ZG
       A(1,6)=-M*YG
       A(2,1)=0.0
       A(2,2)=M-YCOEFF(6)
```

```
      A(2,3)=0.0
      A(2,4)=-(M*ZG+YCOEFF(2)*L)
      A(2,5)=0.0
      A(2,6)=M*XG-YCOEFF(1)*L
      A(3,1)=0.0
      A(3,2)=0.0
      A(3,3)=M-ZCOEFF(5)
      A(3,4)=M*YG
      A(3,5)=-(M*XG+ZCOEFF(1)*L)
      A(3,6)=0.0
      A(4,1)=0.0
      A(4,2) = - ((M * ZG/(L**2)) + KCOEFF(8)/L)
      A(4,3)=M*YG/(L**2)
      A(4,4)=IX-KCOEFF(1)
      A(4,5)=-IXY
      A(4,6)=-(IXZ+KCOEFF(2))
      A(5,1)=M*ZG/(L**2)
      A(5,2)=0.0
      A(5,3)=-((M*XG/(L**2))+MCOEFF(6)/L)
      A(5,4)=-IXY
      A(5,5)=IY-MCOEFF(1)
      A(5,6)=-IYZ
      A(6,1)=-M*YG/(L**2)
      A(6,2)=((M*XG/(L**2))-NCOEFF(6)/L)
      A(6,3)=0.0
      A(6,4)=-(IXZ+NCOEFF(2))
      A(6,5)=-IYZ
      A(6,6)=IZ-NCOEFF(1)
C
C     INITIALIZE TIME, POSITION, AND RATE QUANTITIES
C
      T=0.0D1
      T1=1.
      T2=1.
      T3=1.
      T4=1.
```

```
      T5=1.
      T6=1.
      T7=1.
      T8=1.
      T9=1.
      T10=1.
      T11=1.
      T12=1.
      T13=1.
      T14=1.
      T15=1.
      T16=1.
      T17=1.
      T18=1.
      X=0.001
      Y=0.001
      XDT=0.
      YDT=0.
      ZDT=0.
      ADIFF=0.
      ATHETA=0.
      APHI=0.
      CRS2=100.
      CHECK=0.
C
C     CONVERT ANGLES IN DEGREES TO RADIANS AND KNOTS TO FT/SEC FOR INTERNAL USE.
C
      U0=U
      U0=U0*1.6889
      U=U*1.6889
      V=V*1.6889
      W=W*1.6889
      UDT=UDT*1.6889
      VDT=VDT*1.6889
      WDT=WDT*1.6889
      PHI=PHI/57.2958
```

```
      THETA=THETA/57.2958
      PSI=PSI/57.2958
      COURSE=COURSE/57.2958
      RRATE=RRATE/57.2958
      RUDAMT=RUDAMT/57.2958
      BORATE=BORATE/57.2958
      BOWMAX=BOWMAX/57.2958
      STERAT=STERAT/57.2958
      STERMX=STERMX/57.2958
      MAXANG=MAXANG/57.2958
      ONCRS=ONCRS/57.2958
      NOPICH=NOPICH/57.2958
      IF(T.EQ.0.0D1) GO TO 31
C
C     AS EACH ITERATION BEGINS, INITIALIZE 'AA' MATRIX EQUAL TO 'A' MATRIX.
C     EACH CALLING TO 'FUNC' WILL DESTROY 'AA' MATRIX DURING GAUSSIAN REDUCTION.
C
   30 DO 53 K=1,6
      DO 53 JJ=1,6
      AA(K,JJ)=A(K,JJ)
   53 CONTINUE
C
C     CALL 'FUNC' SUBROUTINE AND CALCULATE ACCELERATIONS AT TIME T + DELTA T.
C
      CALL FUNC
C
C     UPDATE ANGLES AND VELOCITIES
C
      PHI=PHI+(DELT*P)+((DELT**2)/2)*PDT
      THETA=THETA+(DELT*Q)+((DELT**2)/2)*QDT
      PSI=PSI+(DELT*R)+((DELT**2)/2)*RDT
      U=U+DELT*UDT
      V=V+DELT*VDT
      W=W+DELT*WDT
      P=P+DELT*PDT
      Q=Q+DELT*QDT
```

```
      R=R+DELT*RDT
C
C     CALCULATE VELOCITY OF SUBMARINE IN FIXED COORDINATE SYSTEM.
C
      XDT=    ((U*(DCOS(THETA)*DCOS(PSI))+V*(-DSIN(PSI)*DCOS(PHI)+DSIN(P
     1HI)*DSIN(THETA)*DCOS(PSI))+W*(DSIN(PHI)*DSIN(PSI)+DCOS(PHI)*DCOS(P
     2SI)*DSIN(THETA)))*DELT)
      YDT=    ((U*(DCOS(THETA)*DSIN(PSI))+V*(DCOS(PHI)*DCOS(PSI)+DSIN(PH
     1I)*DSIN(THETA)*DSIN(PSI))+W*(-DSIN(PHI)*DCOS(PSI)+DCOS(PHI)*DSIN(T
     2HETA)*DSIN(PSI)))*DELT)
      ZDT=    ((U*(-DSIN(THETA))+V*(DSIN(PHI)*DCOS(THETA))+W*(DCOS(PHI)*
     1DCOS(THETA)))*DELT)
C
C     CALCULATE POSITION OF SUBMARINE (X,Y,Z) AT TIME T + DELTA T.
C
      X=X+DELT*XDT
      Y=Y+DELT*YDT
      Z=Z+DELT*ZDT
      DIFF=Z-ODEPTH
      ADIFF=DABS(DIFF)
      ATHETA=DABS(THETA)
      APSI=DABS(PSI)
      APHI=DABS(PHI)
      PI=3.14159
      CRS1=COURSE-PI
      IF(CRS1.LE.O.) CRS2=DABS(COURSE-PSI)
      IF(CRS1.GT.O.) CRS2=DABS(PI-CRS1+PSI)
C
C     CALL SUBROUTINES TO ADJUST RUDDER AND PLANES AS NECESSARY TO ACHIEVE
C     ORDERED COURSE AND DEPTH
C
      IF((CRS2.GT.ONCRS).OR.(DELR.NE.O.)) CALL RUDDER
   17 IF(ADIFF.GT.NODIFF) CALL DEPTH
      IF(ADIFF.GT.DCRIT) CALL STERN
      IF((ADIFF.LT.DCRIT).AND.(ATHETA.GT.NOPITCH)) CALL STERN
C
```

```
C     CONVERT ANGLES IN RADIANS TO DEGREES AND FT/SEC TO KNOTS FOR OUTPUT
C
   31 PDTO=PDT*57.2958
      QDTO=QDT*57.2958
      RDTO=RDT*57.2958
      DELRO=DELR*57.2958
      DELSO=DELS*57.2958
      DELBO=DELB*57.2958
      PHIO=PHI*57.2958
      UOO=U/1.6889
      VO=V/1.6889
      WO=W/1.6889
      PSIO=PSI*57.2958
      PHIO=PHI*57.2958
      THETAO=THETA*57.2958
      UDTO=UDT/1.6889
      VDTO=VDT/1.6889
      WDTO=WDT/1.6889
      PO=P*57.2958
      QO=Q*57.2958
      RO=R*57.2958
C
C     STORE VALUES THAT CANNOT BE WRITTEN NOW IN MATRIX 'STOW' FOR OUTPUT
C     LATER WITH A NEW HEADING.
C
      STOW(J,1)=T
      STOW(J,2)=PHIO
      STOW(J,3)=THETAO
      STOW(J,4)=PSIO
      STOW(J,5)=PO
      STOW(J,6)=QO
      STOW(J,7)=RO
      STOW(J,8)=PDTO
      STOW(J,9)=QDTO
      STOW(J,10)=RDTO
      STOW(J,11)=DELRO
```

```
      WRITE(6,50)T,X,Y,Z,U00,VO,WO,UDTO,VDTO,WDTO,DELSO,DELBO
   50 FORMAT(3(1X,D10.3),D11.4,8(1XD10.3),/)
C
C     TIME INCREMENT ALGORITHM.
C
   21 T=T+DELT
      J=J+1
C
C     END CALCULATIONS WHEN ORDERED COURSE AND DEPTH IS REACHED
      IF((CRS2.LE.ONCRS).AND.(ADIFF.LT.NODIFF).AND.(ATHETA.LE.NOPICH).AN
     1D.(APHI.LE.1.0)) GO TO 39
C
C
C     END CALCULATIONS WHEN 'ICNT' NUMBER OF ITERATIONS ARE COMPLETE.
C
      IF (ICNT.GE.600) GO TO 39
      ICNT=ICNT+1
      GO TO 30
C
C     WRITE NEW HEADING FOR OUTPUT THAT HAS BEEN STORED IN MATRIX 'STOW'.
C
   39 WRITE(6,48) (ID(I),I=1,40)
   48 FORMAT(1H1,39X,'**************************************************
     1*',/,40X,'*',49X,'*',/,40X,'*    ',40A1,'     *',/,40X,'*',49X,'*'
     2,/,40X,'***************************************************',///)
      WRITE(6,44)
   44 FORMAT(5X,'T',11X,'PHI',6X,'THETA',6X,'PSI',10X,'P',10X,'Q',10X,'R
     *',9X,'PDT',8X,'QDT',8X,'RDT',7X,'DELR')
      WRITE(6,47)
   47 FORMAT (3X,'(SEC)',6X,'(DEG)',6X,'(DEG)',6X,'(DEG)',4X,'(DEG/SEC)'
     1,2X,'(DEG/SEC)',2X,'(DEG/SEC) (DEG/SEC2) (DEG/SEC2) (DEG/SEC2)',4X
     2,'(DEG)')
      WRITE(6,43)((STOW(J,I),I=1,11),J=1,ICNT)
   43 FORMAT(11(1X,D10.3),/)
C
C     CHECK TO SEE IF NEW DATA IS WAITING FOR ANOTHER RUN.
```

```
C
      GO TO 1
  999 STOP
      END
```

```
      SUBROUTINE FUNC
C
      IMPLICIT REAL*8(A-H),REAL*8(L-Z)
      REAL*8 XCOEFF(16),YCOEFF(22),ZCOEFF(22),KCOEFF(17),MCOEFF(23),NCOE
     1FF(22),AA(6,6),E(6),AI(3),BI(3),CI(3),WKAREA(6,6)
      REAL*8 IX,IY,IZ,IXY,IXZ,IYZ
      COMMON /FOUR/ U,V,W,UO,PHI,AI,BI,CI,XCOEFF,YCOEFF,ZCOEFF,KCOEFF,MC
     1OEFF,NCOEFF,L,M,P,RHO,WT,B,XG,YG,ZG,IZ,IX,IY,IXZ,IXY,IYZ,RDT,QDT,P
     2DT,WDT,VDT,UDT,XB,YB,ZB,A1,A2,AA,WKAREA
      COMMON /FIVE/ THETA,Q,DELS,DELB,DELT,PSI,R,DELR
C
C     SUBROUTINE FUNC CALCULATES THE RIGHT HAND SIDE OF THE SIX NON-LINEAR
C     DIFFERENTIAL EQUATIONS.
C
      ETA=0.0D1
C
C     CALCULATE MAGNITUDE OF VECTOR VELOCITY
C
      BIGU=DSQRT(U**2+V**2+W**2)
      ETA=UO/BIGU
      THETAR=THETA
      PHIR=PHI
      VW=DSQRT(V**2+W**2)
C
C     TEST FOR APPROPRIATE THRUST/DRAG COEFFICIENT
C
      IF(ETA.LE.A1) GO TO 10
      IF((ETA.GT.A1).AND.(ETA.LE.A2)) GO TO 11
      IF(ETA.GT.A2) GO TO 12
   10 AII=AI(1)
      BII=BI(1)
      CII=CI(1)
      GO TO 40
   11 AII=AI(2)
      BII-BI(2)
```

```
      CII=CI(2)
      GO TO 40
   12 AII=AI(3)
      BII=BI(3)
      CII=CI(3)
C
C        RIGHT-HAND-SIDE OF LINEAR D.E.'S FOR UDT, VDT, WDT, PDT, QDT, AND RDT.
C
   40 E(1)=(L*(XCOEFF(1)*(Q**2)+XCOEFF(2)*(R**2)+XCOEFF(3)*R*P))+(XCOEFF
     1(5)*V*R+XCOEFF(6)*W*Q)+((1./L)*(XCOEFF(7)*(U**2)+XCOEFF(8)*(V**2)+
     2XCOEFF(9)*(W**2)+AII*(U**2)+BII*U*UO+CII*(UO**2)))+((1./L)*(U**2)*
     3(XCOEFF(10)*(DELR**2)+XCOEFF(11)*(DELS**2)+XCOEFF(12)*(DELB**2)))+
     4((1./L)*(ETA-1.)*(XCOEFF(13)*(V**2)+XCOEFF(14)*(W**2)+XCOEFF(15)*(
     5(DELR*U)**2)+XCOEFF(16)*((DELS*U)**2)))-((2./(RHO*(L**3)))*(WT-B)*
     6DSIN(THETAR))-(M*(W*Q-V*R-XG*((Q**2)+(R**2))+YG*P*Q+ZG*P*R))
C
      VV=V
      IF(VV.EQ.0.0D1) VV=0.1D-15
C
      E(2)=L*(YCOEFF(3)*P*DABS(P)+YCOEFF(4)*P*Q+YCOEFF(5)*Q*R)+(YCOEFF(7
     1)*Q*V+YCOEFF(8)*W*P+YCOEFF(9)*W*R+YCOEFF(10)*U*R+YCOEFF(11)*U*P+YC
     2OEFF(12)*U*DABS(R)*DELR+YCOEFF(13)*(V/DABS(VV))*DABS(VW)*DABS(R)+Y
     3COEFF(19)*U*R*(ETA-1))+((1/L)*(YCOEFF(14)*(U**2)+YCOEFF(15)*U*V+YC
     4OEFF(16)*V*DABS(VW)+YCOEFF(17)*V*W+YCOEFF(18)*(U**2)*DELR))+((2/(R
     5HO*L**3))*(WT-B)*DCOS(THETAR)*DCOS(PHIR))+(((1/L)*(ETA-1))*(YCOEFF
     6(20)*U*V+YCOEFF(21)*V*DABS(VW)+YCOEFF(22)*DELR*(U**2)))-(M*(U*R-W*
     7P-YG*(R**2+P**2)+ZG*Q*R+XG*Q*P))
C
      WW=W
      IF(WW.EQ.0.0D1) WW=0.1D-15
C
      E(3)=(L*(ZCOEFF(2)*P**2+ZCOEFF(3)*(R**2)+ZCOEFF(4)*R*P))+ZCOEFF(6)
     1*V*R+ZCOEFF(7)*V*P+ZCOEFF(8)*U*Q+ZCOEFF(9)*U*DABS(Q)*DELS+ZCOEFF(1
     20)*(W/DABS(WW))*DABS(VW)*DABS(Q)+ZCOEFF(11)*U*Q*(ETA-1)+((1/L)*(ZC
     3OEFF(12)*(U**2)+ZCOEFF(13)*U*W+ZCOEFF(14)*W*DABS(VW)+ZCOEFF(15)*U*
     4DABS(W)+ZCOEFF(16)*DABS(W*VW)+ZCOEFF(17)*(V**2)+ZCOEFF(18)*(U**2)*
```

```
     5DELS+ZCOEFF(19)*(U**2)*DELB))+ ((1/L)*(ETA-1))*(ZCOEFF(20)*U*W+ZCO
     6EFF(21)*W*DABS(VW)+ZCOEFF(22)*DELS*(U**2))+((2/(RHO*L**3))*(WT-B)*
     7DCOS(THETAR)*DCOS(PHIR))-(M*(V*P-U*Q-ZG*((P**2)+(Q**2))+XG*R*P+YG*
     8R*Q))
C
      E(4)=KCOEFF(3)*Q*R+KCOEFF(4)*P*Q+KCOEFF(5)*P*DABS(P)+((1/L)*(KCOEF
     1F(6)*U*P+KCOEFF(7)*U*R+KCOEFF(9)*V*Q+KCOEFF(10)*W*P+KCOEFF(11)*W*R
     2))+((1/L**2)*(KCOEFF(12)*(U**2)+KCOEFF(13)*V*U+KCOEFF(14)*V*DABS(V
     3W)+KCOEFF(15)*V*W+KCOEFF(16)*(U**2)*DELR+KCOEFF(17)*(U**2)*(ETA-1)
     4))+((2/(RHO*L**5))*((YG*WT-YB*B)*DCOS(THETAR)*DCOS(PHIR)-(ZG*WT-ZB
     5*B)*DCOS(THETAR)*DSIN(PHIR)))-((IZ-IY)*Q*R)+(IXZ*Q*P)-((R**2-Q**2)
     6*IYZ)-IXY*P*R-((M*(1/L**2))*(YG*(V*P-U*Q)-ZG*(U*R-W*P)))
C
      E(5)=(MCOEFF(2)*P**2+MCOEFF(3)*(R**2)+MCOEFF(4)*R*P+MCOEFF(5)*Q*DA
     1BS(Q))+((1/L)*(MCOEFF(7)*V*R+MCOEFF(8)*V*P+MCOEFF(9)*U*Q+MCOEFF(10
     2)*U*DABS(Q)*DELS+MCOEFF(11)*DABS(VW)*Q+MCOEFF(20)*U*Q*(ETA-1)))+((
     31/L**2)*(MCOEFF(12)*(U**2)+MCOEFF(13)*U*W+MCOEFF(14)*W*DABS(VW)+MC
     4OEFF(15)*U*DABS(W)+MCOEFF(16)*DABS(W*VW)+MCOEFF(17)*(V**2)+MCOEFF(
     518)*(U**2)*DELS+MCOEFF(19)*(U**2)*DELB))+((2/(RHO*L**5))*((XG*WT-X
     6B*B)*DCOS(THETAR)*DCOS(PHIR)-(ZG*WT-ZB*B)*DSIN(THETAR)))+((1/L**2)
     7*(ETA-1)*(MCOEFF(21)*U*W+MCOEFF(22)*W*DABS(VW)+MCOEFF(23)*DELS*(U*
     8*2)))-(IY-IX)*P*Q+IYZ*R*P-IXY*((Q**2)-(P**2))-IXZ*R*Q-(M*(1/L**2)*
     9(XG*(U*R-W*P)-YG*(W*Q-V*R)))
C
      E(6)=NCOEFF(3)*P*Q+NCOEFF(4)*Q*R+NCOEFF(5)*R*DABS(R)+(1/L)*(NCOEFF
     1(7)*W*R+NCOEFF(8)*W*P+NCOEFF(9)*V*Q+NCOEFF(10)*U*P+NCOEFF(11)*U*R+
     2NCOEFF(12)*U*DABS(R)*DELR+NCOEFF(13)*DABS(VW)*R) +((1/L**2)*(NCOEF
     3F(14)*U**2+NCOEFF(15)*U*V+NCOEFF(16)*V*DABS(VW)+NCOEFF(17)*W*V+NCO
     4EFF(18)*(U**2)*DELR))+((2./(RHO*L**5))*((XG*WT-XB*B)*DCOS(THETAR)*
     5DSIN(PHIR)+(YG*WT-YB*B)*DSIN(THETAR)))+((1./L)*(ETA-1.)*NCOEFF(19)
     6*U*R)+((1./L**2)*(ETA-1.)*(NCOEFF(20)*U*V+NCOEFF(21)*V*DABS(VW)+NC
     7OEFF(22)*DELR*U**2))-((IY-IX)*P*Q)+(IYZ*P*R)-(IXY*(Q**2-P**2))-(IX
     8Z*P*R)-(M*(1./L**2)*(XG*(U*R-W*P)-YG*(W*Q-V*R)))
C
C       THIS CALL PERFORMS THE GAUSSIAN REDUCTION OF THE 'AA' MATRIX.
C       LEQT1F IS A LIBRARY FUNCTION IN SYS5.IMSLIB.
```

```
C
      CALL LEQT1F (AA,1,6,6,E,3,WKAREA,IER)
      RDT=E(6)
      QDT=E(5)
      PDT=E(4)
      WDT=E(3)
      VDT=E(2)
      UDT=E(1)
      RETURN
      END
```

```
      SUBROUTINE RUDDER
C
      IMPLICIT REAL*8(A-H), REAL*8(L-Z)
      REAL*8 K1,K2,K3,K4,K5,K6,K7,K8,K9,K10
      COMMON /THREE/ K9,K10,T5,T6,T11,T12,T17,T18,TLAGR,RRATE,RUDAMT,COU
     1RSE
      COMMON /FIVE/ THETA,Q,DELS,DELB,DELT,PSI,R,DELR
C
      PI=3.14159
      DELTA=RRATE*DELT
C
C     PERFORM A TEST TO DECIDE WHICH WAY TO TURN. DCORS REPRESENTS
C     THE DIFFERENCE BETWEEN THE PRESENT HEADING AND THE ORDERED
C     COURSE.
C
      TEST=COURSE-PI
      IF(TEST.LE.0.) DCORS=COURSE-PSI
      IF(TEST.GT.0.) DCORS=-((PI-TEST)+PSI)
C
C     WHEN THE MODEL IS NEARLY ON COURSE THE RUDDER SHOULD BE
C     BROUGHT AMIDSHIPS AND THE ANGULAR VELOCITY SHOULD BE
C     CLOSELY MONITORED.
C
      CLOSE=.12*TEST
      IF(DABS(DCORS).LE.CLOSE) GO TO 18
C
C      CALCULATED RUDDER DEFLECTION IS PROPORTIONAL TO THE ERROR
C      SIGNAL AND THE ANGULAR VELOCITY.
C
      DEL4=K9*DCORS+K10*R
      IF(DEL4.GT.0.) GO TO 1
C
C      ENSURE TIME LAG HAS PASSED BEFORE RUDDER MOVES.
C
      T5=0.
```

```
      IF(T6.LE.TLAGR) GO TO 3
      IF(DABS(DEL4).LT.DELR) GO TO 5
      DELR=DELR+DELTA
C
C       ENSURE MAX. ORDERED RUDDER DEFLECTION IS NOT EXCEEDED.
C
   14 IF(DELR.GE.RUDAMT) DELR=RUDAMT
      GO TO 99
C
C       ENSURE TIME LAG HAS PASSED BEFORE RUDDER MOVES.
C
    5 T11=0.
      IF(T12.LE.TLAGR) GO TO 7
C
C     AS ERROR SIGNAL GETS SMALLER, RUDDER ANGLE CAN BE REDUCED.
C
      DELTB=DELR-DABS(DEL4)
      IF(DELTB.LT.DELTA) DELR=DELR-DELTB
      IF(DELTB.GT.DELTA) DELR=DELR-DELTA
      GO TO 99
C
C     ENSURE TIME LAG HAS PASSED BEFORE RUDDER MOVES.
C
    1 T6=0.
      IF(T5.LE.TLAGR) GO TO 4
      IF(DABS(DEL4).LT.DABS(DELR)) GO TO 6
      DELR=DELR-DELTA
C
C       ENSURE MAX. ORDERED RUDDER DEFLECTION IS NOT EXCEEDED.
C
   10 IF(DABS(DELR).GE.RUDAMT) DELR=-RUDAMT
      GO TO 99
C
C       ENSURE TIME LAG HAS PASSED BEFORE RUDDER MOVES.
C
    6 T12=0.
```

```
      IF(T11.LE.TLAGR) GO TO 8
C
C      AS ERROR SIGNAL GETS SMALLER, RUDDER ANGLE CAN BE REDUCED.
C
      DELTB=DABS(DELR)-DABS(DEL4)
      IF(DELTB.LT.DELTA) DELR=DELR+DELTB
      IF(DELTB.GT.DELTA) DELR=DELR+DELTA
      GO TO 99
C
C      ALGORITHM FOR SMALL ERROR SIGNALS.
C
   18 IF(R.LT.0.) GO TO 11
      IF(R.GT.0.) GO TO 12
      GO TO 99
C
C      ENSURE TIME LAG HAS PASSED BEFORE RUDDER MOVES.
C
   11 T17=0.
      IF(T18.LE.TLAGR) GO TO 9
      DELR=DELR-DELTA
      IF((DELR.LT.0.).AND.(DCORS.LE.0.)) GO TO 15
      GO TO 10
C
C      ENSURE TIME LAG HAS PASSED BEFORE RUDDER MOVES.
C
   12 T18=0.
      IF(T17.LE.TLAGR) GO TO 13
      DELR=DELR+DELTA
      IF((DELR.GT.0.).AND.(DCORS.GE.0.)) GO TO 15
      GO TO 14
   15 DELR=0.
      GO TO 99
    9 T18=T18+DELT
      GO TO 99
   13 T17=T17+DELT
      GO TO 99
```

```
 3 T6=T6+DELT
   GO TO 99
 4 T5=T5+DELT
   GO TO 99
 7 T12=T12+DELT
   GO TO 99
 8 T11=T11+DELT
99 RETURN
   END
```

```
      SUBROUTINE DEPTH
C
      IMPLICIT REAL*8(A-H), REAL*8(L-Z)
      REAL*8 K1,K2,K3,K4,K5,K6,K7,K8,K9,K10
      COMMON /TWO/ K1,K2,T1,T2,T13,T14,T15,T16,TLAGB,BORATE,BOWMAX,T,CHE
     1CK
      COMMON /FIVE/ THETA,Q,DELS,DELB,DELT,PSI,R,DELR
      COMMON /SIX/ DIFF,ADIFF,ZDT,ATHETA,MAXANG,NOPICH,DCRIT
C
C      CALCULATED DIVE PLANE ANGLE IS PROPORTIONAL TO DEPTH ERROR
C      SIGNAL AND RATE OF CHANGE OF DEPTH.
C
      DEL1=K1*DIFF+K2*ZDT
      ADEL1=DABS(DEL1)
      DELTA=BORATE*DELT
      DELTB=DABS(DELB)-ADEL1
C
C      IF THE ERROR SIGNAL IS SMALL GO TO 18 FOR FINER CONTROL.
C
      IF(ADIFF.LT.DCRIT) GO TO 18
      IF(DEL1.LT.0.) GO TO 4
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
      T2=0.
      IF(T1.LE.TLAGB) GO TO 89
C
C      IF MAX DIVE/ASCENT ANGLE IS EXCEEDED TAKE CORRECTIVE ACTION.
C
    5 IF((ATHETA.GE.MAXANG).AND.(THETA.GE.0.)) GO TO 2
      DELB=DELB+DELTA
      IF(DELB.GT.ADEL1) GO TO 10
C
C      ENSURE MAX PLANE ANGLE IS NOT EXCEEDED.
C
```

```
    9 IF(DELB.GE.BOWMAX) DELB=BOWMAX
C
C      IF MAX DIVE/ASCENT ANGLE IS EXCEEDED WRITE DIAGNOSTIC.
C
      IF(ATHETA.GE.MAXANG) GO TO 6
      GO TO 91
    2 DELB=DELB-DELTA
      GO TO 8
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
   10 T15=0.
      IF(T16.LE.TLAGB) GO TO 88
C
C      AS ERROR SIGNAL GROWS SMALLER, THE PLANE ANGLE CAN BE REDUCED.
C
      IF(DELTB.LE.DELTA) DELB=ADEL1
      IF(DELTB.GT.DELTA) DELB=DELB-DELTA
      GO TO 92
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
    4 T1=0.
      IF(T2.LE.TLAGB) GO TO 90
C
C      IF MAX DIVE/ASCENT ANGLE IS EXCEEDED TAKE CORRECTIVE ACTION.
C
      IF((ATHETA.GE.MAXANG).AND.(THETA.LE.0.)) GO TO 3
      DELB=DELB-DELTA
      IF(DABS(DELB).GT.ADEL1) GO TO 11
C
C      ENSURE MAX PLANE ANGLE IS NOT EXCEEDED.
C
    8 IF(DABS(DELB).GE.BOWMAX) DELB=-BOWMAX
C
C      IF MAX DIVE/ASCENT ANGLE IS EXCEEDED WRITE DIAGNOSTIC.
C
```

```
      IF(ATHETA.GE.MAXANG) GO TO 6
      GO TO 91
    3 DELB=DELB+DELTA
      GO TO 9
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
   11 T16=0.
      IF(T15.LE.TLAGB) GO TO 87
C
C      AS ERROR SIGNAL GROWS SMALLER, THE PLANE ANGLE CAN BE REDUCED.
C
      IF(DELTB.LE.DELTA) DELB=-ADEL1
      IF(DELTB.GT.DELTA) DELB=DELB+DELTA
      GO TO 92
C
C      DIAGNOSTIC WRITE STATEMENT
C
    6 WRITE(6,7)T
    7 FORMAT(/,20X,'***** EXCEEDED MAX DIVE/ASCENT ANGLE AT TIME =',F10
     *.4,'  *****',/,20X,'*****  STANDARD FIXUP TAKEN *****')
      GO TO 91
C
C      CONTROL FOR SMALL ERROR SIGNALS.
C
   18 IF((ZDT.LT.0.).AND.(DIFF.GT.0.).AND.(DELB.EQ.0.)) RETURN
      IF((ZDT.GT.0.).AND.(DIFF.LT.0.).AND.(DELB.EQ.0.)) RETURN
      IF((ZDT.GT.0.).AND.(DIFF.GT.0.).AND.(DELB.LE.4.)) GO TO 14
      IF((ZDT.GT.0.).AND.(DIFF.LT.0.).AND.(DELB.NE.0.)) GO TO 12
      IF((ZDT.GT.0.).AND.(DIFF.GT.0.).AND.(DELB.GT.4.))  RETURN
      IF((ZDT.LT.0.).AND.(DIFF.GT.0.).AND.(DELB.NE.0.)) GO TO 12
      IF((ZDT.LT.0.).AND.(DIFF.LT.0.).AND.(DELB.GE.-4.)) GO TO 13
      IF((ZDT.LT.0.).AND.(DIFF.LT.0.).AND.(DELB.LT.-4.)) RETURN
   12 IF(DELB.LT.0.) GO TO 22
      CHECK=2.
```

```
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
   13 T13=0.
      IF(T14.LE.TLAGB) GO TO 17
      IF(CHECK.EQ.2.) GO TO 23
      IF(DELB.GT.-6.) GO TO 20
      GO TO 91
   23 IF((DELB-DELTA) .LT.0.) GO TO 15
   20 DELB=DELB-DELTA
      GO TO 92
   22 CHECK=1.
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
   14 T14=0.
      IF(T13.LE.TLAGB) GO TO 16
      IF(CHECK.EQ.1.) GO TO 21
      IF(DELB.LT.6.) GO TO 19
      GO TO 91
   21 IF((DELB+DELTA).GT.0.) GO TO 15
   19 DELB=DELB+DELTA
      GO TO 92
   15 DELB=0.
      GO TO 91
   16 T13=T13+DELT
      GO TO 91
   17 T14=T14+DELT
      GO TO 91
   87 T15=T15+DELT
      GO TO 91
   88 T16=T16+DELT
      GO TO 91
   89 T1=T1+DELT
      GO TO 91
   90 T2=T2+DELT
```

```
   GO TO 91
92 IF(DELB.LT.-BOWMAX) DELB=-BOWMAX
   IF(DELB.GT.BOWMAX) DELB=BOWMAX
91 RETURN
   END
```

```
      SUBROUTINE STERN
C
      IMPLICIT REAL*8(A-H), REAL*8(L-Z)
      REAL*8 K1,K2,K3,K4,K5,K6,K7,K8,K9,K10
      COMMON /ONE/ K3,K4,K5,K6,K7,K8,T3,T4,T7,T8,T9,T10,TLAGS,STERAT,STE
     1RMX
      COMMON /FIVE/ THETA,Q,DELS,DELB,DELT,PSI,R,DELR
      COMMON /SIX/ DIFF,ADIFF,ZDT,ATHETA,MAXANG,NOPICH,DCRIT
C
C      CALCULATED PLANE ANGLE IS PROPORTIONAL TO DEPTH ERROR, RATE OF
C      CHANGE OF DEPTH, PITCH ANGLE, AND RATE OF CHANGE OF PITCH.
C
      DEL3=K5*DIFF+K6*ZDT+K7*THETA+K8*Q
      ADEL3=DABS(DEL3)
      DELTA=STERAT*DELT
      DELTB=DABS(DELS)-ADEL3
C
C      FOR SMALL DEPTH ERROR AND LARGE PITCH ERROR GO TO 18
C
      IF((ADIFF.LT.DCRIT).AND.(ATHETA.GT.NOPICH)) GO TO 18
      IF(DEL3.GT.0.) GO TO 2
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
      T3=0.
      IF(T4.LE.TLAGS) GO TO 8
C
C      IF MAX DIVE/ASCENT ANGLE IS EXCEEDED, TAKE CORRECTIVE ACTION.
C
      IF((ATHETA.GE.MAXANG).AND.(THETA.LE.0.)) GO TO 5
      IF(DELS.GT.ADEL3) GO TO 6
      DELS=DELS+DELTA
C
C      ENSURE MAX ORDERED PLANE ANGLE IS NOT EXCEEDED.
C
```

```
    1 IF(DELS.GE.STERMX) DELS=STERMX
      GO TO 99
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
    2 T4=0.
      IF(T3.LE.TLAGS) GO TO 9
C
C      IF MAX DIVE/ASCENT ANGLE IS EXCEEDED, TAKE CORRECTIVE ACTION.
C
      IF((ATHETA.GE.MAXANG).AND.(THETA.GE.0.)) GO TO 4
      IF(DABS(DELS).GT.ADEL3) GO TO 7
      DELS=DELS-DELTA
C
C      ENSURE MAX ORDERED PLANE ANGLE IS NOT EXCEEDED.
C
    3 IF(DABS(DELS).GE.STERMX) DELS=-STERMX
      GO TO 99
    4 DELS=DELS+DELTA
      GO TO 1
    5 DELS=DELS-DELTA
      GO TO 3
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
    6 T8=0.
      IF(T7.LE.TLAGS) GO TO 10
C
C      AS ERROR SIGNAL GROWS SMALLER, THE PLANE ANGLE CAN BE REDUCED.
C
      IF(DELTB.LE.DELTA) DELS=ADEL3
      IF(DELTB.GT.DELTA) DELS=DELS-DELTA
      GO TO 99
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
```

```
    7 T7=0.
      IF(T8.LE.TLAGS) GO TO 11
C
C      AS ERROR SIGNAL GROWS SMALLER, THE PLANE ANGLE CAN BE REDUCED.
C
      IF(DELTB.LE.DELTA) DELS=-ADEL3
      IF(DELTB.GT.DELTA) DELS=DELS+DELTA
      GO TO 99
    8 T4=T4+DELT
      GO TO 99
    9 T3=T3+DELT
      GO TO 99
   10 T7=T7+DELT
      GO TO 99
   11 T8=T8+DELT
      GO TO 99
   12 IF(DELS.LT.0.) GO TO 14
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
   13 T9=0.
      IF(T10.LE.TLAGS) GO TO 17
      IF((DELS-DELTA).LT.0.) GO TO 15
   20 DELS=DELS-DELTA
      GO TO 99
C
C      ENSURE TIME LAG HAS PASSED BEFORE PLANE MOVES.
C
   14 T10=0.
      IF(T9.LE.TLAGS) GO TO 16
      IF((DELS+DELTA).GT.0.) GO TO 15
   19 DELS=DELS+DELTA
      GO TO 99
   15 DELS=0.
      GO TO 99
   16 T9=T9+DELT
```

```
      GO TO 99
   17 T10=T10+DELT
      GO TO 99
C
C       CONTROL TO CORRECT PITCH ANGLE.
C
   18 IF((THETA.LT.0.).AND.(Q.GT.0.).AND.(DELS.EQ.0.)) RETURN
      IF((THETA.GT.0.).AND.(Q.LT.0.).AND.(DELS.EQ.0.)) RETURN
      IF((THETA.LT.0.).AND.(Q.LT.0.).AND.(DELS.LT.0.)) RETURN
      IF((THETA.GT.0.).AND.(Q.GT.0.).AND.(DELS.GT.0.)) RETURN
      IF((THETA.LT.0.).AND.(Q.GT.0.).AND.(DELS.NE.0.)) GO TO 12
      IF((THETA.GT.0.).AND.(Q.LT.0.).AND.(DELS.NE.0.)) GO TO 12
      IF((THETA.LT.0.).AND.(Q.LT.0.).AND.(DELS.GT.0.)) GO TO 13
      IF((THETA.GT.0.).AND.(Q.GT.0.).AND.(DELS.LT.0.)) GO TO 14
      IF((THETA.GT.0.).AND.(Q.GT.0.).AND.(DELS.EQ.0.)) GO TO 19
      IF((THETA.LT.0.).AND.(Q.LT.0.).AND.(DELS.EQ.0.)) GO TO 20
   99 RETURN
      END
```

B.3 LEQT1F

A	- input matrix of dimension N by N containing the coefficient matrix of the equation AX = B. On output, A is replaced by the LU decomposition of a rowwise permutation of A.
M	- number of right-hand sides. (input)
N	- order of A and number of rows in B. (input)
IA	- number of rows in the dimension statement for A and B in the calling program. (input)
B	- input matrix of dimension N by M containing right-hand sides of the equation AX = B. On output, the N by M solution X replaces B.
IDGT	- input option. If IDGT is greater than 0 the elements of A and B are assumed to be correct to IDGT Decimal digits and the routine performs an accuracy test. If IDGT equals zero, the accuracy test is bypassed.
WKAREA	- work area of dimension greater than or equal to N.
IER	- error parameter terminal error = 128 + N N = 1 indicates that A is algorithmically singular. Warning error = 32 + N. N = 2 indicates that the accuracy test failed. The computed solution may be in error by more than can be accounted for by the uncertainty of the data.

CALL LEQT1F (A, M, N, IA, B, IDGT, WKAREA, IER)

184696

Thesis
H1753 Hammond
c.1

Development of a six degree of freedom motion simulation model for use in submarine design analysis.